A SCHEDULING PROGRAM FOR A MULTIPROGRAMMED COMPUTER

by

HERBERT JAY BOROCK

B.S., Columbia University

(1963)

SUBMITTED IN PARTIAL FULFILLMENT

OF THE REQUIREMENTS FOR THE

DEGREE OF MASTER OF

SCIENCE

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June, 1967

Signature of Author..
Alfred P. Sloan School of Management, May 19, 1967

Certified by....
Thesis Supervisor

Accepted by..
Chairman, Departmental Committee on Graduate Students

RESTRICTED DISTRIBUTION

A SCHEDULING PROGRAM FOR A MULTIPROGRAMMED COMPUTER

by

HERBERT JAY BOROCK

Submitted to the Alfred P. Sloan School of Management on May 19, 1967 in partial fulfillment of the requirements for the degree of master of science.

ABSTRACT

This thesis describes a scheduling program for an IBM System/360 computer operating in a multiprogramming environment provided by the IBM System/360 Disk Operating System. In this environment up to three programs compete for the use of the computer's resources.

A scheduling rule is developed for computer center operations managers in the form of a computer program. The program saves substantial amounts of computer time since it provides more efficient schedules for the multiprogramming environment than could be provided by the operations manager without using the program. Input to the program consists of statements describing the computer, the installation, and the programs to be scheduled. Output from the program consists of a suggested schedule, showing program sequence and clock time for the scheduled period, which is a week.

The problem of scheduling the computer is solved by dividing the problem into two smaller scheduling problems--a job shop scheduling problem and a flow shop scheduling problem. In the job shop scheduling problem, three programs compete for the use of the computer's central processing unit and input/output channels. Given each program's characteristics, the time required to complete each program is calculated by simulating the operation of the computer.

In the flow shop scheduling problem, each program can use up to three areas of computer storage for program execution. This is an n-job, 3-machine sequencing problem, where each program is a job and each area of computer storage is a machine. Previous algorithms for the flow shop scheduling problem are used to sequence the programs for the class of problems in which two machines cannot operate on the same job in parallel. New branch and bound algorithms are developed for the class of three-machine problems in which two of the machines can operate on the same job in parallel.

Thesis Advisor: Donald C. Carroll

Title: Associate Professor of Management

May 19, 1967

Professor Edward N. Hartley
Secretary of the Faculty
Massachusetts Institute of Technology
Cambridge, Massachusetts 02139

Dear Professor Hartley:

In accordance with the requirements for graduation, I herewith submit a thesis entitled "A Scheduling Program for a Multiprogrammed Computer."

I want to take this opportunity to thank the many people who assisted me in my research.

The National Dairy Products Corporation provided the financial assistance for this work. Paul A. Strassmann, Director of Systems and Procedures, suggested the topic. Robert I. Benjamin, Manager of Installation Support offered many helpful suggestions.

My thesis committee, consisting of Professors Donald C. Carroll and John F. Pierce, were a constant source of encouragement and were always available for advice.

Computer time was supplied by the M.I.T. Computation Center and the Computer Facility of the Alfred P. Sloan School of Management.

Mrs. Anita Ciminero typed the final draft.

Sincerely,

Herbert Jay Borock

Herbert Jay Borock

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

Chapter 1

INTRODUCTION

Scheduling a multiprogrammed computer has been recognized for some time as analogous to scheduling the work in a factory. However, little practical work has been done to assist computer center operations managers in scheduling programs run on a multiprogrammed computer. Part of the reason for this is that until recently multiprogrammed computers were used in only large computer installations. Further, when researchers tried to attack the problem they dealt with hypothetical computers. With the introduction of the IBM System/360, multiprogramming is being extended to smaller sized installations. The typical business computer installation is now faced with the problem of scheduling work on a multiprogrammed computer.

In most cases the installation chooses a manufacturer-supplied programming system which provides no scheduling assistance, except for job-to-job transitions. All the programming system does is free the operator from some routine work so that he can devote more time to other routine work. No assistance is given in actually scheduling the jobs, i.e., deciding which job sequence results in the most efficient use of the computer. The purpose of this research is to provide the computer center operations manager with guidance in this area. Using the scheduling program developed here, the manager will be able to obtain good or "near" optimal schedules.

Previous Research

Lynch[1] described the operation of a university computing center. In that environment, the majority of the programs use less than 5 minutes of computing time. Less than 5% of the programs use any input/output devices other than for card input and printer output. All of the programs are limited by computing time, rather than input/output time. Use of the computer is allocated to different groups of individuals on a priority basis.

Hutchinson[2] described the simulation of an entire computing center, including accounting-in and accounting-out operations, and the use of a tape library. The scope of his study is limited to large aerospace-manufacturing computing centers. He varied manpower levels and equipment to determine the most efficient center. His measure of service was the amount of time to process jobs.

McKenney[3] developed a system which defined computer jobs so that they could be recursively segmented into independent subjobs. An independent subjob is defined by the source of its inputs, the definition

---

[1] W. C. Lynch, "Description of a High Capacity, Fast Turnaround University Computing Center," Communications of the ACM, (IX, February, 1966), 117-123.

[2] George K. Hutchinson, "A Computer Center Simulation Project," Communications of the ACM, (VIII, September, 1965), 559-568.

[3] James L. McKenney, "Simultaneous Processing of Jobs on an Electronic Computer," Management Science, (VIII, April, 1962), 344-354.

of its outputs, and a relation between its inputs and outputs. He concluded that computer programs should not be restricted by arbitrary sequence constraints, so that they could be segmented into independent subjobs, which could be processed by different computer components in parallel.

Heller[4] used a linear graph technique to indicate the use of each computer component and the precedence relationships among subprograms. He formulated the problem; he did not solve it.

Schwartz[5] studied the feasibility of restructuring programs to take advantage of multiple facility computers. He attempted to develop a procedure by which the computer could restructure the programs and then schedule them. He was concerned with the network properties of computer programs and the derivation of a precedence matrix.

Codd[6,7] developed an algorithm for a multiple facility system. He considered two kinds of facilities--those that were time-shared, and those that were space-shared. The purpose of his algorithm was to schedule a set of programs so that they could be executed in a minimum amount of time.

---

[4]J. Heller, "Sequencing Aspects of Multiprogramming, "Journal of the Association for Computing Machinery, (VIII, July, 1961), 426-439.

[5]Eugene S. Schwartz, "An Automatic Sequencing Procedure with Application to Parallel Programming,"Journal of the Association for Computing Machinery (VIII, October, 1961), 513-537.

[6]E. F. Codd, "Multiprogram Scheduling: Parts 1 and 2. Introduction and theory," Communications of the ACM, (III, June, 1960), 347-350.

[7]E. F. Codd, "Multiprogram Scheduling: Parts 3 and 4. Scheduling Algorithm and External Constraints," Communications of the ACM, (III, July, 1960), 413-418.

The Problem

The computer we are concerned with is used by business concerns for useful work on a continuing basis. The work in these installations consists of high volume input and output. It is not uncommon for a single program to include hours of card reading and printing. Since these input/output operations are slow relative to other processing, systems programs have been developed to provide a computer environment where one job's processing can be overlapped with other jobs' input/output operations. In this way the work load can be balanced so that one job is not delayed by excessive input, say, and another is not delayed by excessive processing.

The computer environment we are concerned with consists of three programs being multiprogrammed together. One program is a processing program, and the other two programs are input/output programs. The majority of the jobs on the computer use the processing program area of storage and at least one of the input/output programs.

Most jobs enter the computer installation at the beginning of one day and must be completed at the end of that day. The remainder must be completed within the same week they enter the installation. Other than having different due dates, there is no priority among jobs.

This environment poses two problems. First, how much time does each program take to run on the computer? The time is easy to calculate if there is only one program using the computer, since the input for each program consists of a known number of transactions, and the processing time per transaction and input/output time per transaction are known. However, in this environment three programs are sharing the computer, thus complicating the timing problem.

Second, given that we know how long each program takes, how do we schedule the programs to make efficient use of the computer? The scheduling problem is easy if each program uses the computer only once. However, in this environment each program may use the computer three times--once for processing, and twice for input/output.

The Solution

The two problems are solved separately. The timing problem is treated as a job shop scheduling problem where three programs compete for the use of the computer's central processing unit and input/output channels. Given each program's characteristics (e.g., number of machine instructions per input record), the time required to complete each program is calculated by simulating the operation of the computer.

The scheduling problem itself is treated as a flow shop scheduling problem where each of three areas of computer storage is analogous to a machine. Each time a program uses the computer it uses a different machine. This is the familiar n-job, 3-machine sequencing problem. However, there is a substantial difference. Both input/output areas may perform the same function, i.e., they may be identical machines, or at the very least, a single program may use them in parallel. New branch and bound algorithms are developed for the parallel operation cases.

The Scheduling Program as a Management Control Device

The scheduling rule developed here is provided to computer center operations managers in the form of a computer program. The program is written for the IBM System/360, the same computer that is in the computer center. The program saves substantial amounts of computer time since it provides more efficient schedules for a multiprogramming environment than could be provided by the operations manager without using the program.

Input to the program consists of statements describing the computer, the installation, and the programs to be scheduled. The program descriptions, which are provided by the programmers who design the programs, include such things as the usage of various input/output devices.

Output from the program consists of a suggested schedule, showing program sequence and clock time for the scheduled period, which is a week. The operations manager has the opportunity to revise the schedule after which the schedule becomes his plan for processing the work in the computer center. The actual performance of the computer center and its manager can then be compared to the planned performance, so that more efficient and effective performance can be produced.

Chapter 2

SCHEDULING ENVIRONMENT

A scheduling environment consists of resources, jobs, and a scheduling rule. The resources and jobs in this scheduling environment are the components of a computer and the programs run on the computer, respectively. They are described in this chapter. The scheduling rule is described in Chapter 6.

The Computer Components[1]

The computer for which this scheduling program is written is the IBM System/360. A System/360 consists of main storage, a central processing unit (CPU), up to six selector channels, one multiplexor channel, and input/output devices attached to the channels. Programs and data reside in main storage. The CPU is used to execute a program's instructions and to transfer data between the input/output units and main storage via the channels.

The channels are used to relieve the CPU of the task of communicating directly with the input/output units and permit instruction execution to occur concurrently with data transfer. A selector channel transfers data to or from a single input/output device at a time and is capable of handling high speed input/output devices. The multiplexor channel can operate in either of two modes: multiplex mode for low-speed devices, and burst mode for high speed devices.

---

[1]For a more detailed discussion see IBM System/360 Principles of Operation (Poughkeepsie: IBM Corporation, 1964).

In the multiplex mode, a large number of low-speed input/output devices time-share the channel's single data path, and the channel simultaneously transfers data between all the devices and main storage. In the burst mode, a single high-speed device uses the channel's data path, and that device does not relinquish the channel until its data transfer operation is complete.

The CPU can concurrently execute a machine instruction, transfer data to or from a single input/output device on each of the selector channels, and transfer data to or from either a single high-speed device on the multiplexor channel (burst mode) or multiple low-speed devices on the multiplexor channel (multiplex mode).

System/360 is designed to be multiprogrammed. A multiprogrammed System/360 is one in which two or more programs reside in main storage and compete for the use of the CPU, the selector channels, and possibly the multiplexor channel. They compete for the use of the multiplexor channel if at least one program wants to use that channel in the burst mode; otherwise, all programs can use the multiplexor channel at the same time.

A programming system is used to control the operation of the computer and to allocate the use of the CPU and the channels among the programs. The programming system for which this scheduling program is written is the IBM System/360 Disk Operating System (DOS/360).

DOS/360 provides a multiprogramming environment where main storage is divided into three areas, each of which can contain a single program.[2] Programs are loaded into one of the areas of main

---

[2]For a more detailed discussion see IBM System/360 Disk Operating System: System Control and Service Programs (Endicott: IBM Corporation, 1966).

storage (called the background area) by DOS/360 as they are needed, without operator intervention. Programs are loaded into the other two areas of main storage (called foreground area one and foreground area two) by the operator.

When they compete for the CPU and the channels, the program in foreground area one has the highest priority, the program in foreground area two has the next highest priority, and the program in the background area has the lowest priority. Control of the CPU is given to the highest priority program which is not waiting for the completion of an input/output operation. That program can then issue machine instructions and initiate input/output operations.

If the program issues a multiplex mode input/output operation, and the multiplexor channel is operating in the multiplex mode, the input/output operation is executed immediately. If the program issues a burst mode input/output operation and the multiplexor channel is in use, or a multiplex mode input/output operation and the multiplexor channel is operating in burst mode, or an input/output operation to a selector channel and the channel is in use, then the input/output operation is executed when the channel finishes executing the previous input/output operation issued to it.

If the program cannot execute any more machine instructions until a particular input/output operation is complete, the program relinquishes control of the CPU to a lower priority program which is not waiting for the completion of an input/output operation. When the input/output operation is complete, the higher priority program seizes control of the CPU from the lower priority program.

The Programs

The programs run on the computer are either foreground programs or background programs, depending on the area of main storage in which they are executed.

The foreground programs are used to transfer data between a low-speed device on the multiplexor channel and a high-speed device. Since the operation of the low-speed device takes much longer than the instructions used to initiate the input/output operations on both devices, only a small portion of the total program time is used to execute instructions on the CPU. A somewhat longer time is used to transmit data over the channel to which the high-speed device is attached. Most of the time is used to transmit data over the multiplexor channel in the multiplex mode. This permits the background program to use the CPU and the channels while the foreground programs are waiting for the completion of input/output operations on the multiplexor channel.

Foreground programs may be used as input or output transcription programs in conjunction with background programs, or they may operate independently of the background programs. Both foreground programs may be used for the same function (e.g., output transcription) at the same time.

For example, a typical background program includes many low-speed input/output operations on the multiplexor channel for input transcription (e.g., card reading) and output transcription (e.g., printing). If the background program actually performed these operations, a substantial amount of the program time would consist of waiting for the completion of these operations as is the case with the foreground programs.

Instead of performing card reading and printing directly with the background program, a foreground program can transfer the input data to a high-speed device, the background program can then read the data from that device and put its output data on another high-speed device, which another foreground program can then transfer to the printer.

In this example, while one foreground program is reading input for a subsequent background program, another foreground program is printing the output for a previous background program, and a background program is being executed using data previously read by the first foreground program and supplying data which will subsequently be printed by the second foreground program. Depending on the relative volumes of input and output, both foreground programs may be used for either input or output, one foreground program may be used for either input or output while the other is doing independent work, or both foreground programs may be doing independent work.

Some sets of background programs represent a single accounting unit of processing in terms of DOS/360. The programs in each set must be executed in sequence, with no other programs intervening. These background program sets may operate in conjunction with foreground programs as described above.

The Weekly Schedule

In the installations we are concerned with, most of the programs enter the computer installation at the beginning of one day and must be completed by the end of that day. Other programs have different turnaround times, but all programs must be completed during the same calendar week that they enter the computer installation. Thus, each week a static schedule must be produced for a group of foreground programs and background programs. Given descriptions of the computer (CPU, channel, and input/output device performance characteristics), the installation (computer configuration, number and normal use of foreground areas, and work week), and the programs (execution times, input/output volumes, device usage, and possible foreground program usage), the object is to produce an optimal or near optimal schedule for the week. During the week, conditions in the computer center may change, such that the original schedule is no longer the best schedule: new programs are added to the schedule; the operators make errors and programs have to be rerun; an input/output device goes down. In that case, a new schedule would have to be produced, given the descriptions of the computer, the installation, and the programs when the change occurs.

The problem of scheduling programs in this environment can be divided into two smaller scheduling problems--a job shop scheduling problem and a flow shop scheduling problem. The scheduling rule is in the form of a computer program which is run on the same computer for which the schedule is produced. As long as the computer is capable of performing useful work, then no matter what kinds of changes occur during the week, a new schedule can be produced.

The Job Shop Scheduling Problem

Each of the three programs being multiprogrammed can require the use of up to nine computer components: CPU for instruction execution, up to six selector channels for data transfer, multiplexor channel for data transfer in burst mode, and multiplexor channel for data transfer in multiplex mode. At any one instant in time, only one program can use the CPU for executing instructions, only one program can use each of the selector channels to transfer data between main storage and an input/output device, and either only one program can use the multiplexor channel to transfer data in burst mode or all programs can use the multiplexor channel to transfer data in multiplex mode.

Given a program's description, it is easy to calculate the time to complete the program, if it is the only program on the computer. All one has to do is calculate its time on the CPU and on each of the channels, and then choose the longest of these times as the total program time, since the operations on these computer components are overlapped.

However, when more than one program is being run at the same time, the scheduling problem is more complex. More than one program is assigned to the same resource (multiplexor channel in multiplex mode) at the same time. More than one resource (e.g., CPU and a channel) is assigned to the same program at the same time. A program uses a resource (CPU or any channel) more than once.

It would be unwise to keep track of all the changes in resource assignment in order to calculate the time for each program in the multiprogramming environment. To do so would require simulating each machine instruction and input/output operation. Such a simulation would require more time to execute on a computer than would the schedule itself.

The approach taken here is first to calculate the total CPU time and channel time for each of the programs, assuming that each program is the only one on the computer. Then, using the DOS/360 program priorities, and the fact that the CPU time of one program can be overlapped with the channel time of that program and the other two programs, new program times are calculated. The simulation method used to calculate these times is described in Chapter 3.

The Flow Shop Scheduling Problem

Given the program times calculated in solution to the job shop scheduling problem, the problem of scheduling the week's work on the computer reduces to a flow shop scheduling program.

In the flow shop scheduling problem, each area of main storage represents a machine. The program times represent operation times on these machines. Jobs consist of from one to three operations, with no more than one operation on each machine. A three operation job consists of a background program run in conjunction with two foreground programs which read the background program's input and/or print its output.

If both foreground programs are used in conjunction with the background programs, then we have an n-job, 3-machine sequencing problem. If only one foreground program is used in conjunction with the background programs, then we have an n-job, 2-machine sequencing problem, since the other foreground program can always operate in parallel with both the first foreground program and the background program.

There are four distinct sequencing problems:

1. The n-job, 2-machine sequencing problem (background programs plus a foreground program for either input or output).
2. The n-job, 3-machine sequencing problem (background programs plus one foreground program each for input and output).
3. The n-job, 3-machine sequencing problem (background programs plus two foreground programs for input).
4. The n-job, 3-machine sequencing problem (background programs plus two foreground programs for output).

Algorithms which provide optimal solutions to problems 1 and 2 have been developed previously. They are described in Chapter 4. As far as I know, algorithms which provide optimal solutions to problems 3 and 4 have not been developed previously. Branch and bound algorithms for problems 3 and 4 are described in Chapter 5.

Chapter 3

SIMULATION METHOD FOR THE JOB SHOP SCHEDULING PROBLEM

Up to three programs may be multiprogrammed in the environment being simulated. For each program, the total CPU time and the total channel time for each channel are calculated, assuming that each program is the only one on the computer. The maximum of these times for each program is the total program time for that program in the non-multiprogrammed environment, since the operations on these computer components are overlapped.[1] The times on these computer components are then expressed as fractions of the total program time for each program.

Two Component Example

Suppose there are three programs--program 1, program 2, and program 3, each of which uses only two computer components--the CPU and the multiplexor channel in the multiplex mode. When it is the only program using the computer, program 1 uses the CPU for 10 minutes and the multiplexor channel for 100 minutes; program 2 uses the CPU for 10 minutes and the multiplexor channel for 50 minutes; and program 3 uses the CPU for 45 minutes and the multiplexor channel for 150 minutes. Then in the non-multiprogrammed environment, the total program times for programs 1, 2, and 3 are 100 minutes, 50 minutes, and

---

[1]This is true for most of the programs we are concerned with. Some programs, however, do have non-overlapped operations. The scheduling rule developed here provides for such programs by including the percentage of non-overlapped operations in the input to the scheduling program.

150 minutes, respectively, since the time on the multiplexor channel is the maximum time for each program.

Program 1 uses the CPU 10/100 or .1 of the total time it uses the computer, and program 1 uses the multiplexor channel 100/100 or 1.0 of the total time it uses the computer. Similarly, program 2 uses the CPU .2 of the time, and the multiplexor channel 1.0 of time; and program 3 uses the CPU .3 of the time, and the multiplexor channel 1.0 of the time. These times are shown in Figure 1.

	Programs 1	2	3
CPU/Total time	.1	.2	.3
Multiplexor Channel/Total time	1.0	1.0	1.0

Figure 1. Component use as a fraction of total program time

We now know how long each program uses each computer component during one time unit, when the programs are not multiprogrammed together. We have to calculate how long each program uses each computer component during one time unit, when the programs are multiprogrammed together. During one time unit, however, a program uses each computer component in the same proportion that it uses the component over the total program time. Therefore, we have to calculate how long each program uses the computer during each time unit.

Let CPU/MPX(i) represent the fraction of the total program time that program i uses both the CPU and the multiplexor channel. Let MPX(i) represent the fraction of the total program time that program i uses only the multiplexor channel. Then CPU/MPX(1)=.1 and MPX(1)=.9, since program 1's total program time is 100 minutes, and during 10 minutes of this time it uses both the CPU and the multiplexor channel; and during the remaining 90 minutes it uses only the multiplexor channel. Similarly, CPU/MPX(2)=.2 and MPX(2)=.8; and CPU/MPX(3)=.3 and MPX(3)=.7. These fractions are shown in Figure 2.

| CPU/MPX(1)=.1 | MPX(1)=.9 |

| CPU/MPX(2)=.2 | MPX(2)=.8 |

| CPU/MPX(3)=.3 | MPX(3)=.7 |

Figure 2. Component use as a fraction of one time unit

Now suppose that the three programs are multiprogrammed together, and that program 1 has the highest priority, program 2 has the next highest priority, and program 3 has the lowest priority.

During one time unit, program 1 uses the computer for the entire time unit, since it can seize control of the CPU whenever it needs it. Program 2 can use the CPU only when program 1 is not using it, and program 3 can use the CPU only when programs 1 and 2 are not using it.

When program 1 is using both the CPU and the multiplexor channel, program 2 can also use the multiplexor channel, since multiplex mode operations on the multiplexor channel can be overlapped with each other. During any time period, program 2 uses the multiplexor channel alone for a fraction MPX(2)=.8 of the time period. During the time period

CPU/MPX(1)=.1 time unit, program 2 uses the multiplexor channel for (.8)*(.1)=.08 time unit.

During the time period MPX(1)=.9 time unit, program 2 can use either the multiplexor channel alone, or both the multiplexor channel and the CPU. In any case, program 2 uses the computer during the entire .9 time unit. The total time that program 2 uses the computer during one time unit is .08+.9=.98 time unit, as shown in Figure 3.

CPU/MPX(1)=.1	MPX(1)=.9
MPX(2)=.08	MPX(2)+CPU/MPX(2)=.9

Figure 3. Total computer use by program 2 during one time unit

However, during any time period, program 2 uses each computer component in the same proportion that it uses the component over the total program time. Specifically, CPU/MPX(2)=.2 and MPX(2)=.8. But program 2 is active for only .98 time unit, so in one time unit, CPU/MPX(2)=(.2)*(.98)=.196 time unit, and MPX(2)=(.8)*(.98)=.784 time unit. These times are shown in Figure 4. Note that .704+.08= .784=MPX(2).

CPU/MPX(1)=1	MPX(1)=.9	
MPX(2)=.08	MPX(2)+CPU/MPX(2)=.9	
MPX(2)=.08	CPU/MPX(2)=.196	MPX(2)=.704

Figure 4. Component use by program 2 during one time unit.

Let CPU/MPX(1,2) represent the fraction of a time unit when both the CPU and the multiplexor channel are being used by either program 1, program 2, or both programs 1 and 2. Let MPX(1,2) represent the fraction of a time unit when only the multiplexor channel is being used by either program 1, program 2, or both programs 1 and 2. Figure 5 shows these fractions as derived from Figure 4. CPU/MPX(1,2)=.296 and MPX(1,2) =.704.

CPU/MPX(1)=.1		MPX(1)=.9	
MPX(2)=.08	CPU/MPX(2)=.196		MPX(2)=.704
CPU/MPX(1,2)=.296			MPX(1,2)=.704

Figure 5. Component use by programs 1 and 2 during one time unit

---

When programs 1 and 2 are using both the CPU and the multiplexor channel, program 3 can also use the multiplexor channel, since multiplex mode operations can be overlapped with each other. During any time period, program 3 uses the multiplexor channel alone for a fraction MPX(3)=.7 of the time period. During the time period CPU/MPX(1,2)=.296 time unit, program 2 uses the multiplexor channel for (.7)*(.296)=.2072 time unit.

During the time period MPX(1,2)=.704 time unit, program 3 can use either the multiplexor channel alone, or both the multiplexor channel and the CPU. In any case, program 3 uses the computer during the entire .704 time unit. The total time that program 3 uses the computer during one time unit is .2072+.704=.9112 time unit, as shown in Figure 6.

CPU/MPX(1,2)=.296	MPX(1,2)=.704	
MPX(3)=.2072	MPX(3)+CPU/MPX(3)=.704	

Figure 6. Total computer use by program 3 during one time unit

However, during any time period, program 3 uses each computer component in the same proportion that it uses the component over the total program time. Specifically, CPU/MPX(3)=.3 and MPX(3)=.7. But program 3 is active for only .9112 time unit, so in one time unit, CPU/MPX(3)=(.3)*(.9112)=.27336 time unit and MPX(3)=(.7)*(.9112)= .63784 time unit. These times are shown in Figure 7. Note that .43064+.2072=.63784=MPX(3).

CPU/MPX(1,2)=.296	MPX(1,2)=.704	
MPX(3)=.2072	MPX(3)+CPU/MPX(3)=.704	
MPX(3)=.2072	CPU/MPX(3)=.27336	MPX(3)=.43064

Figure 7. Component use by program 3 during one time unit

To summarize, during 1 time unit, program 1 uses the computer for 1.0 time unit, program 2 uses the computer for .98 time unit, and program 3 uses the computer for .9112 time unit. These fractions are divided into the total program times calculated for the non-multiprogramming environment to determine how long each program takes in the multiprogramming environment. In this example, the total time for program 1 is 100/1.0=100 minutes; the total time for program 2 is 50/.98=51 minutes; and the total time for program 3 is 150/.9112= 165 minutes.

Use with the Flow Shop Scheduling Problem

The program with the minimum total time (in this case program 2) is the first program to complete processing. Normally it will be replaced by a new program, and the process of calculating multiprogramming times will continue. However, if the foreground programs are being used in conjunction with the background programs, the same foreground programs will always be using the computer, and the only programs that will replace one another are the background programs. Since the flow shop scheduling problem exists only when at least one of the foreground programs is used in conjunction with the background programs, we only have to consider that case now. (We still have to simulate in the other cases, but only when we have found a place in the schedule for the program in question. However, we need to simulate the programs in the flow shop scheduling problem before we place them in the schedule.)

When the foreground programs are being used in conjunction with the background programs, the fractions calculated for each foreground program will always be the same, since the programs do not change. The only time fractions change is when a new background program is loaded into storage. Given a set of programs which constitute a job, with one program to be executed in each of the three areas of storage, the foreground program times can be calculated by dividing the non-multiprogrammed program times by the relevant fractions, which had previously been computed. The background program time can be calculated only after a new fraction, unique to this background program, is calculated.

General Case

Programs usually use more than two computer components. For example, a foreground transcription program uses the CPU, the multiplexor channel, and a selector channel. The method for calculating multiprogrammed program times is extended in this section to cover the general case where each program uses more than 2 computer components.

Let program 1, program 2, and program 3 be foreground program 1, foreground program 2, and a background program, respectively. Let $c_1/c_2/\ldots/c_n(i)$ represent the fraction of the total program time that program i uses computer components c_1, c_2, ..., c_n at the same time. Let $c_1/c_2/\ldots/c_n(1,2)$ represent the fraction of a time unit when all the computer components c_1, c_2, ..., c_n are being used by either program 1, program 2, or both programs 1 and 2.

Then the total time that program 1 uses the computer during one time unit is 1.0 time unit, since program 1 can seize control of the CPU and the selector channels whenever it needs them. Program 2 can use the CPU and the selector channels only when program 1 is not using them, but it can use some of them while program 1 is using the others. Program 3 can use the CPU and the selector channels only when programs 1 and 2 are not using them, but it can use some of them while programs 1 and 2 are using the others.

The total time that program 2 uses the computer is calculated as follows:

1. Consider each time period $c_1/c_2/\ldots/c_n(1)$, the fraction of the total time that program 1 uses components c_1, c_2, ..., c_n at the same time. Calculate the time that program 2 uses the computer during each of these time periods.
2. Sum the times calculated for program 2.

The total time that program 3 uses the computer is calculated as follows:

1. Consider each time period $c_1/c_2/\ldots/c_n(1,2)$, the fraction of the total time that either program 1, program 2, or both programs 1 and 2 use components c_1, c_2, ..., c_n at the same time. Calculate the time that program 3 uses the computer during each of the time periods.
2. Sum the times calculated for program 3.

Example

For example, in the non-multiprogramming environment, suppose program 1 uses the CPU for 10 minutes, selector channel 1 for 30 minutes, and the multiplexor channel in the multiplex mode for 100 minutes; program 2 uses the CPU for 10 minutes, selector channel 2 for 30 minutes, and the multiplexor channel in the multiplex mode for 100 minutes; and program 3 uses the CPU for 20 minutes, selector channel 1 for 50 minutes, and selector channel 2 for 100 minutes. Then in the non-multiprogramming environment, the total program times for programs 1, 2, and 3 are all 100 minutes, since the time on the multiplexor channel is the maximum time for programs 1 and 2, and the time on selector channel 2 is the maximum time for program 3.

Program 1 uses the CPU 10/100 or .1 of the total time it uses the computer, program 1 uses selector channel 1 30/100 or .3 of the total time it uses the computer, and program 1 uses the multiplexor channel 100/100 or 1.0 of the total time it uses the computer. Similarly, program 2 uses the CPU .1 of the time, selector channel 2 .3 of the time, and the multiplexor channel 1.0 of the time; and program 3 uses the CPU .2 of the time, selector channel 1 .5 of the time, and selector channel 2 1.0 of the time. These times are shown in Figure 8.

	Programs		
	1	2	3
CPU/Total Time	.1	.1	.2
Selector Channel 1/Total Time	.3	--	.5
Selector Channel 2/Total Time	--	.3	1.0
Multiplexor Channel/Total Time	1.0	1.0	--

Figure 8. Component use as a fraction of total program time

Let CPU, S1, S2, and MPX represent the CPU, selector channel 1, selector channel 2, and the multiplexor channel, respectively. Then, CPU/S1/MPX(1)=.1, S1/MPX(1)=.2, and MPX(1)=.7, since program 1's total program time is 100 minutes, and during 10 minutes of this time it uses the CPU, selector channel 1, and the multiplexor channel together; during 20 minutes of this time it uses both selector channel 1 and the multiplexor channel; and during 70 minutes of this time it uses only the multiplexor channel. Similarly, CPU/S2/MPX(2)=.1, S2/

MPX(2)=.2, and MPX(2)=.7; and CPU/S1/S2(3)=.2, S1/S2(3)=.3, and S2(3)=.5. These fractions are shown in Figure 9.

CPU/S1/MPX(1)=.1	S1/MPX(1)=.2	MPX(1)=.7
CPU/S2/MPX(2)=.1	S2/MPX(2)=.2	MPX(2)=.7
CPU/S1/S2(3)=.2	S1/S2(3)=.3	S2(3)=.5

Figure 9. Component use as a fraction of one time unit

When program 1 is using the CPU, selector channel 1, and the multiplexor channel together, program 2 can use either the multiplexor channel alone or both the multiplexor channel and selector channel 2. During any time period, program 2 uses the multiplexor channel alone for a fraction MPX(2)=.7 of the time period. During the time period CPU/S1/MPX(1)=.1 time unit, program 2 uses the multiplexor channel for (.7)*(.1)=.07 time unit.

During any time period, program 2 uses both the multiplexor channel and selector channel 2 for a fraction S2/MPX(2)=.2 of the time period. During the time period CPU/S1/MPX(1)=.1 time unit, program 2 uses both the multiplexor channel and selector channel 2 for (.2)*(.1)=.02 time unit.

During the time periods S1/MPX(1)=.2 time unit and MPX(1)=.7 time unit, program 2 can use either the multiplexor channel alone, both the multiplexor channel and selector channel 2, or the multiplexor channel, selector channel 2, and the CPU together. In any case, program 2 uses the computer during the entire .2+.7=.9 time unit. The total time that program 2 uses the computer during one time unit is .07+.02+.9= .99 time unit, as shown in Figure 10.

CPU/S1/MPX(1)=.1		S1/MPX(1)=.2	MPX(1)=.7
MPX(2)=.07	S2/MPX(2)=.02	MPX(2)+S2/MPX(2)+CPU/S2/MPX(2)=.9	

Figure 10. Total computer use by program 2 during one time unit

However, during any time period, program 2 uses each computer component in the same proportion that it uses the component over the total program time. Specifically, CPU/S2/MPX(2)=.1, S2/MPX(2)=.2, and MPX(2)=.7. But program 2 is active for only .99 time unit, so in one time unit, CPU/S2/MPX(2)=(.1)*(.99)=.099 time unit, S2/MPX(2)= (.2)*(.99)=.198 time unit, and MPX(2)=(.7)*(.99)=.693 time unit. These times are shown in Figure 11. Note that .02+.04+.138=.198= S2/MPX(2), .07+.138+.485=.693=MPX(2), and .022+.077=.099=CPU/S1/ MPX(2).

Now CPU/S1/MPX(1,2)=.08, CPU/S1/S2/MPX(1,2)=.042, S1/S2/MPX (1,2)=.04, S1/MPX(1,2)=.138, CPU/S2/MPX(1,2)=.077, S2/MPX(1,2)= .138, and MPX(1,2)=.485. Figure 12 shows these times as derived from Figure 11.

When programs 1 and 2 are using both the multiplexor channel and selector channel 1, program 3 can use selector channel 2. During

CPU/S1/MPX(1)=.1 | S1/MPX(1)=.2

MPX(2)=.07 | S2/MPX(2)=.02 | MPX(2)+S2/MPX(2)+CPU/S2/MPX(2)=.2

MPX(2)=.07 | S2/MPX(2)=.02 | CPU/S2/MPX(2)=.022 | S2/MPX(2)=.04 | MPX(2)=.138

MPX(1)=.7

MPX(2)+S2/MPX(2)+CPU/S2/MPX(2)=.7

CPU/S2/MPX(2)=.077 | S2/MPX(2)=.138 | MPX(2)=.485

Figure 11. Component use by program 2 during one unit of time.

CPU/S1/MPX(1)=.1 | S1/MPX(1)=.2

MPX(2)=.07 | S2/MPX(2)=.02 | CPU/S2/MPX(2)=.022 | S2/MPX(2)=.04 | MPX(2)=.138

CPU/S1/MPX(1,2)=.08 | CPU/S1/S2/MPX(1,2)=.042 | S1/S2/MPX(1,2)=.04 | S1/MPX(1,2)=.138

MPX(1)=.7

CPU/S2/MPX(2)=.077 | S2/MPX(2)=.138 | MPX(2)=.485

CPU/S2/MPX(1,2)=.077 | S2/MPX(1,2)=.138 | MPX(1,2)=.485

Figure 12. Component use by program 1 and 2 during one time unit.

any time period, program 3 uses selector channel 2 alone for a fraction S2(3)=.5 of the time period. During the time period S1/MPX(1,2)=.138 time unit, program 3 uses selector channel 2 for (.5)*(.138)=.069 time unit. Similarly, during the time period CPU/S1/MPX(1,2)=.08 time unit, program 3 uses selector channel 2 for (.5)*(.08)=.04 time unit.

During the time period MPX(1,2)=.485 time unit, program 3 can use either selector channel 2 alone, both selector channel 2 and selector channel 1, or selector channel 2, selector channel 1, and the CPU together. In any case, program 3 can use the computer during the entire .485 time unit. Program 3 cannot use the computer during any of the remaining time periods that programs 1 and 2 are using the computer, since during those time periods selector channel 2 is in use, and whenever program 3 uses the computer it must use selector channel 2. Therefore, the total time that program 3 uses the computer during one time unit is .069+.04+ .485=.594 time unit, as shown in Figure 13.

However, during any time period, program 3 uses each computer component in the same proportion that it uses the component over the total program time. Specifically, CPU/S1/S2(3)=.2, S1/S2(3)=.3, and S2(3)=.5. But program 3 is active for only .594 time unit, so in one time unit, CPU/S1/S2(3)=(.2)*(.594)=.1188 time unit, S1/S2(3)= (.3)*(.594)=.1782 time unit, and S2(3)=(.5)*(594)=.297 time unit. These times are shown in Figure 14. Note that .069+.04+.188+.297= .594=S2(3).

| CPU/S1/MPX(1,2)=.08 | CPU/S1/S2/MPX(1,2)=.042 | S1/S2/MPX(1,2)=.04 | S1/MPX(1,2)=.138 |
| S2(3)=.04 | | | S2(3)=.069 |

| CPU/S2/MPX(1,2)=.077 | S2/MPX(1,2)=.138 | MPX(1,2)=.485 |
| | | S2(3)+S1/S2(3)+CPU/S1/S2(3)=.485 |

Figure 13. Total computer use by program 3 during one time unit.

CPU/S1/MPX(1,2)=.08	CPU/S1/S2/MPX(1,2)=.042	S1/S2/MPX(1,2)=.04	S1/MPX(1,2)=.138
S2(3)=.04			S2(3)=.069
S2(3)=.04			S2(3)=.069

CPU/S2/MPX(1,2)=.077	S2/MPX(1,2)=.138	MPX(1,2)=.485
		S2(3)+S1/S2(3)+CPU/S1/S2(3)=.485
		S2(3)=.188 | S1/S2(3)=.1782 | CPU/S1/S2(3)=.1188

Figure 14. Component use by program 3 during one time unit.

Chapter 4

PREVIOUS ALGORITHMS FOR THE FLOW SHOP SCHEDULING PROBLEM

Johnson[1] was the first to try to develop optimal sequencing rules for the two- and three-machine scheduling problems. He was able to develop a rule for the two-machine problem, but he was unable to develop one for the three-machine problem, except in special cases.

The three-mcahine problem remained unsolved, except for enumeration, until two separate studies produced branch and bound algorithms which lead to optimal schedules. The branch and bound technique was first used by Little, Murty, Sweeney, and Karel[2] for solving the traveling salesman problem. The branch and bound technique consists of breaking a problem up into smaller and smaller subsets and calculating a lower bound for the best alternative therein. The bounds guide the partitioning of the subsets and eventually identify an optimum. The subsets are represented as nodes of a tree and the process of partitioning as a branching of the tree.[3]

---

[1] S. M. Johnson, "Optimal Two- and Three-Stage Production Schedules with Setup Times Included," Chapter 2 John F. Muth and Gerald L. Thompson, editors, Industrial Scheduling, (Englewood Cliffs: Prentice-Hall, 1963), 13-20.

[2] John D. C. Little, Katta G. Murty, Dura W. Sweeney, and Caroline Karel, "An Algorithm for the Traveling Salesman Problem," Operations Research, (XI, November-December, 1963), 972-989.

[3] Ibid., p. 974.

Ignall and Schrage[4] and Lomnicki[5] showed how the branch and bound technique could be used to solve the three-machine flow shop scheduling problem. They developed identical branch and bound techniques. Ignall and Schrage's notation is used here.

The N-Job, 2-Machine Sequencing Problem

Johnson's algorithm for the 2-machine problem is as follows:[6]

1. Let a_i be the operation time for the ith job on the first machine, and b_i the time on the second machine. List the a's and b's in two vertical columns.
2. Scan all the operation times for the shortest one.
3. If it is for the first machine, place the corresponding job first.
4. If it is for the second machine, place the corresponding job last.
5. Remove both operation times for the job from the list.
6. Repeat steps 1-5 until all jobs are sequenced.

---

[4]Edward Ignall and Linus Schrage, "Application of the Branch and Bound Technique to Some Flow-Shop Scheduling Problems," Operations Research, (XIII, May-June, 1965), 400-412.

[5]Z. A. Lomnicki, "A Branch-and-Bound Algorithm for the Exact Solution of the Three-machine Scheduling Problem," Operations Research Quarterly, (XVI, March, 1965), 89-100.

[6]Johnson, pp. 16-17

For example, consider the following:

i	a_i	b_i
1	13	3
2	7	12
3	26	9
4	2	6

Then the optimal sequence is (4, 2, 3, 1), and the total operation time is 51.

The N-Job, 3-Machine Sequencing Problem

Ignall and Schrage's algorithm for the 3-machine problem is as follows:[7]

1. Let J_r be a sequence of size r out of a total of n jobs. Let TIMEA(J_r), TIMEB(J_r), and TIMEC(J_r) be the times when machines A, B, and C, respectively, complete processing the last of the r jobs. Let a_i, b_i, and c_i be the processing times of the ith job on machines A, B, and C, respectively. Let $\overline{J_r}$ be the set of n-r jobs that are not in the sequence J_r. Then a lower bound for the sequence J_r is:

$$LB(J_r) = \max \begin{bmatrix} \mathrm{TIMEA}(J_r) + \sum_{\overline{J_r}} a_i + \min_{\overline{J_r}} (b_i + c_i) \\ \mathrm{TIMEB}(J_r) + \sum_{\overline{J_r}} b_i + \min_{\overline{J_r}} c_i \\ \mathrm{TIMEC}(J_r) + \sum_{\overline{J_r}} c_i \end{bmatrix}$$

[7] Ignall and Schrage, pp. 401-403.

2. Assign the lower bound $LB(J_r)$ to the node in a tree which corresponds to the sequence J_r. Keep the lower bounds in a list.

3. Start the tree with a single node which represents a null sequence.

4. Remove the first node from the list.

5. Create a new node for every job that the just removed node has not yet scheduled.

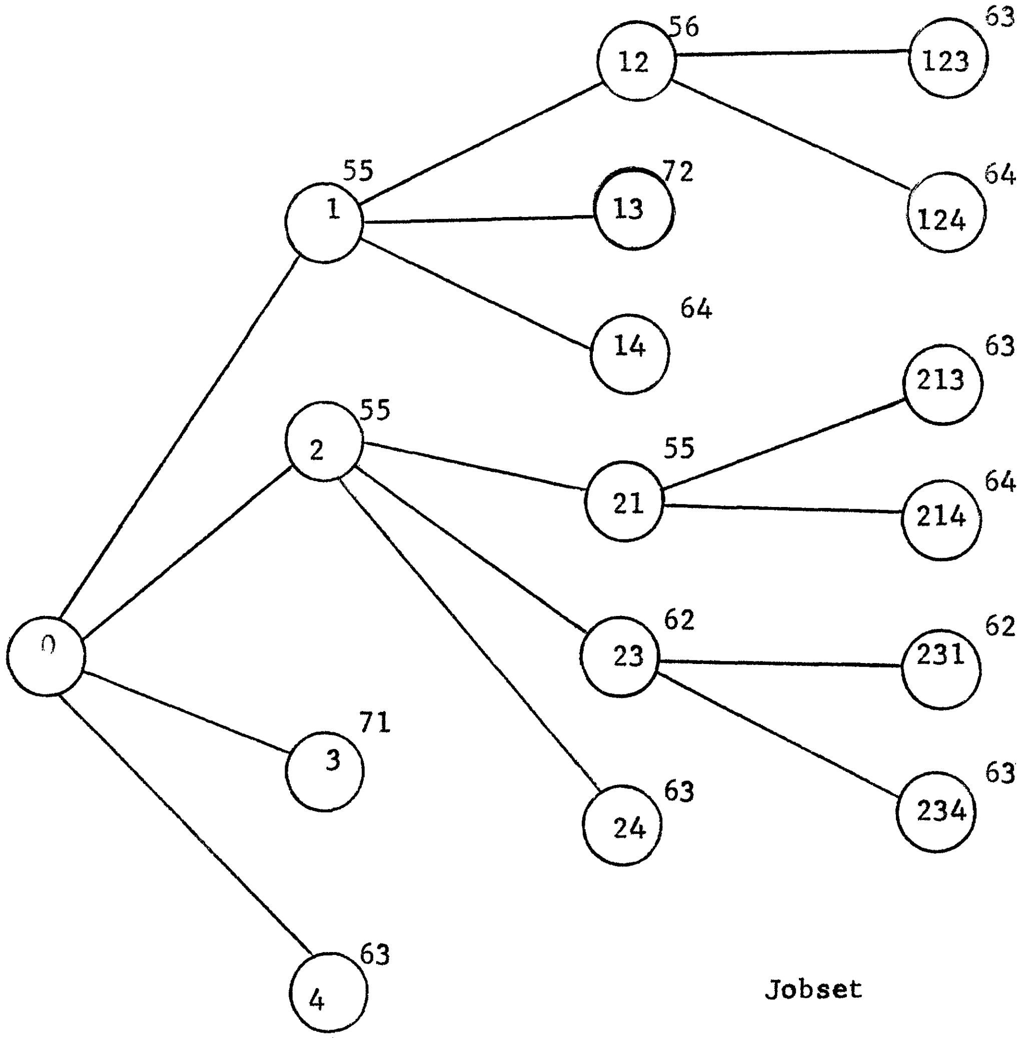
6. Compute the lower bounds for the new nodes and insert them ranked on the list.

7. If a node that has scheduled all jobs is first on the list, that node's sequence is an optimal one.

8. If an optimal sequence has not been found, go to step 4.

Consider the example shown in Figure 15. The lower bound for node 1 is 55, since that is the maximum of 55 (13+35+7), 31 (3+27+1), and 36 (12+24). Nodes 1 and 2 are at the top of the list initially, since the only other nodes are 3 and 4, which have higher lower bounds. Suppose 1 is ahead of 2 on the list. Then node 1 is removed from the list, and nodes 12, 13, and 14 are created, and their lower bounds are computed.

All of the newly created nodes have higher lower bounds than node 2, so node 2 is the next node removed from the list. Nodes 21, 23, and 24 are created, and their lower bounds are computed. Node 21 goes to the top of the list, since it has the smallest lower bound. Node 21 is then removed from the list and nodes 213 and 214 are created.



Jobset

i	1	2	3	4
a_i	13	7	26	2
b_i	3	12	9	6
c_i	12	16	7	1

An optimal sequence is 2-3-1-4

Figure 15. Tree structure for 4-job, 3-machine problem.

Their lower bounds are greater than that of node 12, which is removed from the list. Nodes 123 and 124 are created and are put into the list behind node 23. Node 23 is removed from the list and nodes 231 and 234 are created. Node 231 has a lower bound than any other node in the list. Since node 231 has scheduled all jobs and is first on the list, that node's sequence is an optimal one.

Use of the Algorithms in Scheduling the Computer

Johnson's algorithm and Ignall and Schrage's algorithm are used to produce optimal daily schedules for the following main storage configurations:

1. Johnson's algorithm is used when one foreground program is used in conjunction with the background programs, and the other foreground program does independent work. If the foreground program is used to read the background program's input, then the foreground program is machine A and the background program is machine B. If the foreground program is used to produce the background program's output, then the background program is machine A and the foreground program is machine B.
2. Ignall and Schrage's algorithm is used when one foreground program is used in conjunction with the background program for input, and the other foreground program is used in conjunction with the background program for output. The input foreground program is machine A, the background program is machine B, and the output foreground program is machine C.

Chapter 5

BRANCH AND BOUND ALGORITHMS FOR THE N-JOB, 3-MACHINE, 2-PARALLEL-OPERATION FLOW SHOP SCHEDULING PROBLEM

The algorithms in Chapter 4 can be used to produce optimal daily schedules for many of the storage configurations we are concerned with. However, up to now no algorithm has been developed to produce optimal sequences for the class of three-machine problems in which two of the machines can operate on the same job in parallel. This is a very common problem in the context of scheduling the computer described here. Often both foreground programs are used for background program output, or both are used for background program input.

There are three separate cases for both the input and output problems. The foreground programs may be using different input/output devices (e.g., a card reader and a paper tape reader on input, or a printer and a plotter on output). Both foreground programs may be using identical devices (e.g., two 1000 card per minute card readers on input, or two 1100 line per minute printers on output). The foreground programs may be performing the same operation, but on different input/output device models (e.g., one 1000 card per minute card reader and one 500 card per minute card reader, or one 1100 line per minute printer and one 600 line per minute printer).

Using Ignall and Schrage's notation, the problems can be stated as follows:

Parallel Operations On Input

Let J_r be a sequence of size r out of a total of n jobs. Let $TIMEA(J_r)$, $TIMEA^*(J_r)$, and $TIMEB(J_r)$ be the times when machines A, A*, and B, respectively, complete processing the last of the r jobs. Let a_i, a_i^*, and b_i be the processing times of the ith job on machines A, A*, and B, respectively. Let $\overline{J_r}$ be the set of n-r jobs that are not in the sequence J_r. Then a lower bound for the sequence J_r is:

$$LB(J_r)=\max\left[\begin{array}{l}\max\left[\begin{array}{l} TIMEA(J_r)+\sum_{\overline{J_r}} a_i+\min_{\overline{J_r}} b_i \\ \\ TIMEA^*(J_r)+\sum_{\overline{J_r}} a_i^*+\min_{\overline{J_r}} b_i\end{array}\right] \\ \\ TIMEB(J_r)+\sum_{\overline{J_r}} b_i\end{array}\right]$$

Since $TIMEA(J_r)+\sum_{\overline{J_r}} a_i=\sum a_i$ and $TIMEA^*(J_r)+\sum a_i^*=\sum a_i^*$, the problem

can be expressed in the following form:

$$LB(J_r)=\max\left[\begin{array}{l}\max\left[\begin{array}{l} \sum a_i+\min_{\overline{J_r}} b_i \\ \\ \sum a_i^*+\min_{\overline{J_r}} b_i\end{array}\right] \\ \\ TIMEB(J_r)+\sum_{\overline{J_r}} b_i\end{array}\right]$$

Parallel Operations on Output

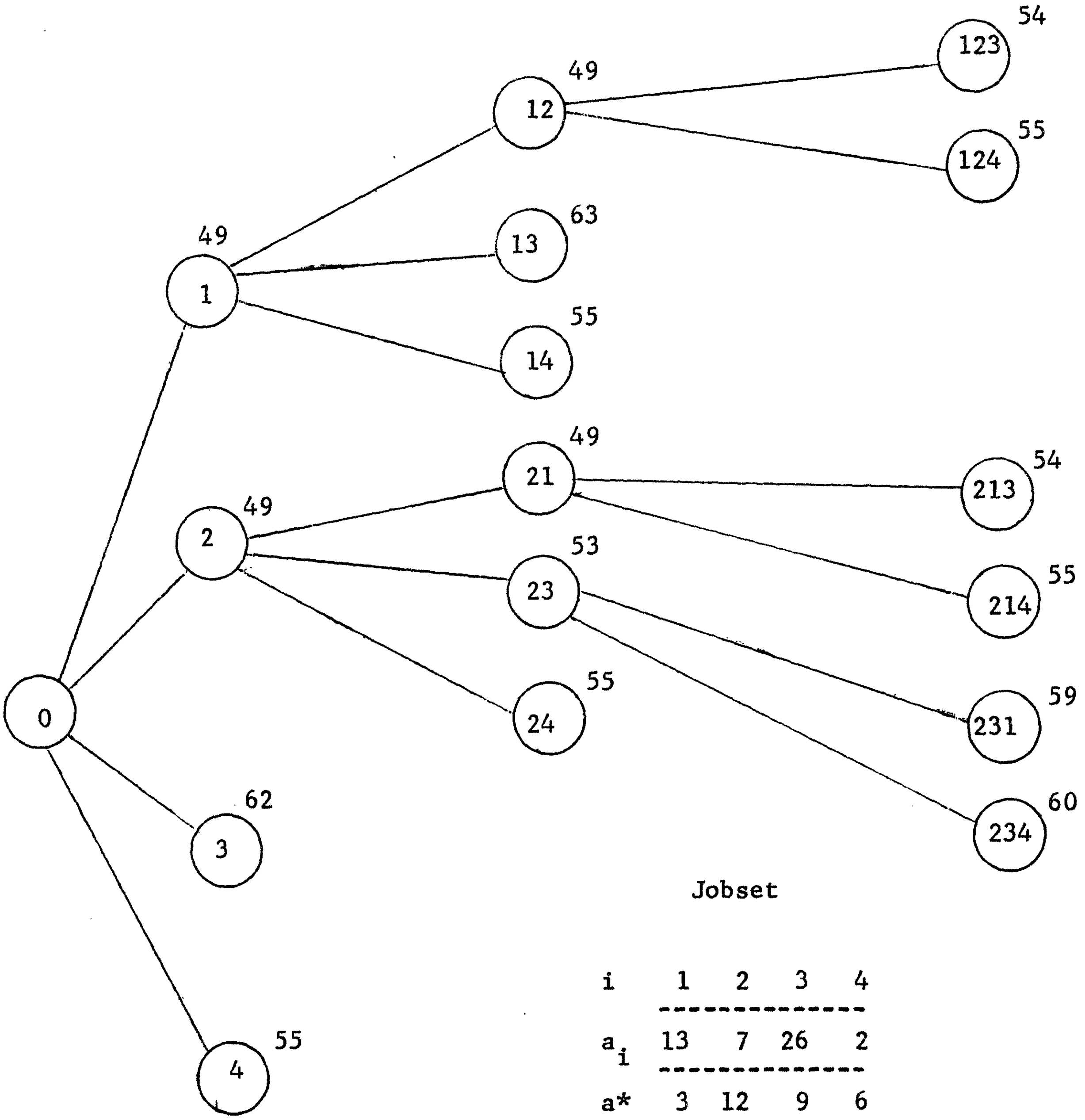
Let J_r be a sequence of size r out of a total of n jobs. Let $TIMEA(J_r)$, $TIMEB(J_r)$, and $TIMEB*(J_r)$ be the times when machines A, B, and B*, respectively, complete processing the last of the r jobs. Let a_i, b_i, and b_i^* be the processing times of the ith job on machines A, B, and B*, respectively. Let $\overline{J_r}$ be the set of n-r jobs that are not in the sequence J_r. Then a lower bound for the sequence J_r is:

$$LB(J_r)=\max\begin{bmatrix} TIMEA(J_r)+\sum_{\overline{J_r}} a_i+\min\begin{bmatrix}\min_{\overline{J_r}} b_i, & b_i > b_i^* \\ \min_{\overline{J_r}} b_i^*, & b_i^* > b_i\end{bmatrix} \\ \max\begin{bmatrix} TIMEB(J_r)+\sum_{\overline{J_r}} b_i \\ TIMEB*(Jr)+\sum_{\overline{J_r}} b_i^* \end{bmatrix}\end{bmatrix}$$

As in the previous problem, $TIMEA(J_r)+\sum_{\overline{J_r}} a_i=\sum a_i$

Case 1: Parallel Operations on Input, Different Operations

Sum the a_i's and the a_i^*'s and choose the larger sum. Choose lower bounds as described above. Follow the algorithm in Chapter 4 for assigning lower bounds to nodes, discarding nodes, and creating new nodes. When a node which has scheduled all jobs is at the top of the list, an optimal solution has been found. Figure 16 illustrates the use of this algorithm.



Jobset

i	1	2	3	4
a_i	13	7	26	2
a^*_i	3	12	9	6
b_i	12	16	7	1

An optimal sequence is 2 - 1 - 3 - 4

Figure 16. Tree structure for 4-job, 3-machine problem, where 2 different operations are performed on different machines in parallel before the third operation.

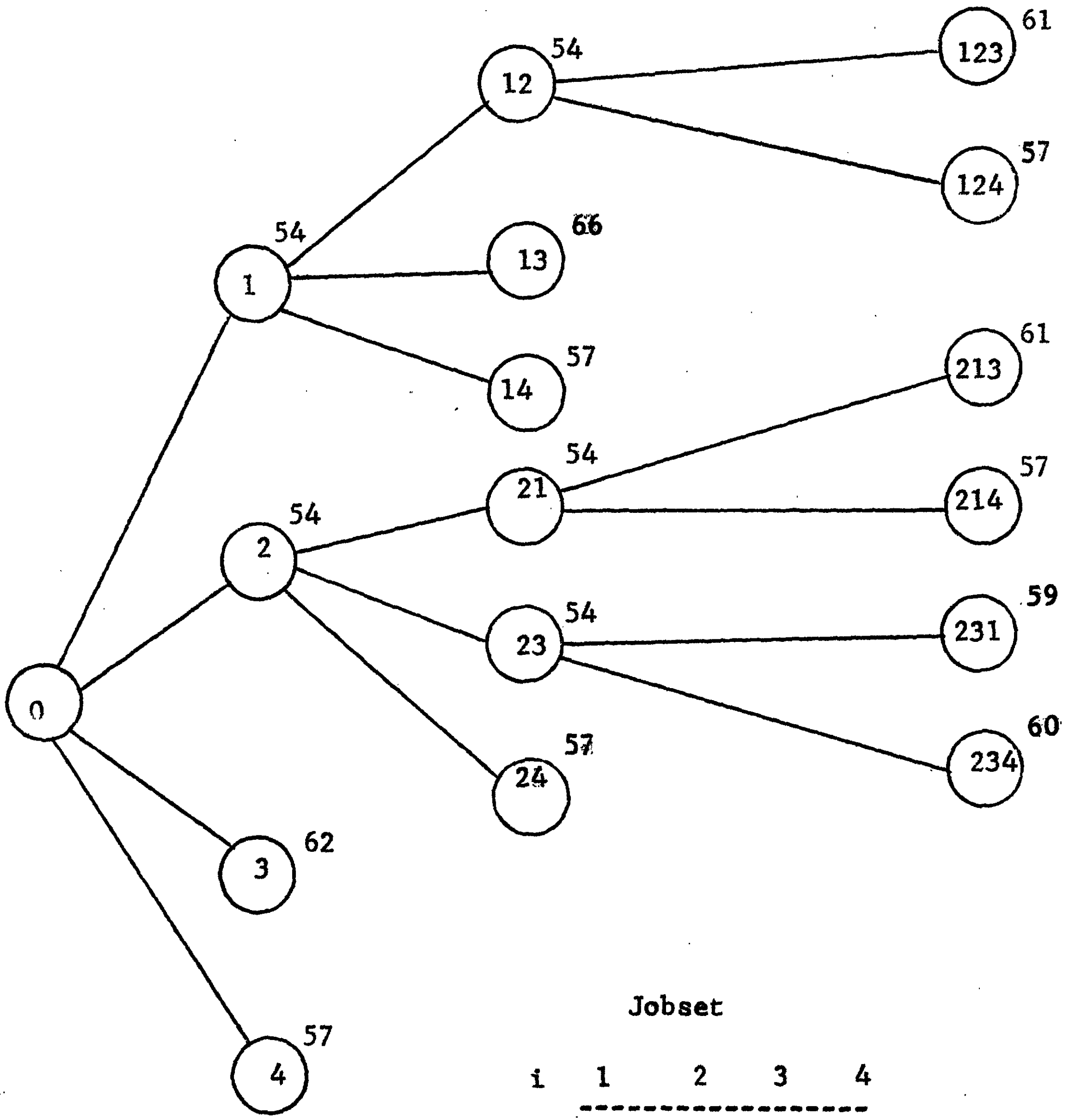
Case 2: Parallel Operations on Output, Different Operations

Sum the a_i's. From the operation times for the remaining jobs, add to the sum the minimum of the following: the minimum b_i, where $b_i > b^*_i$, or the minimum b^*_i, where $b^*_i > b_i$. Choose lower bounds as described above. Follow the algorithm in Chapter 4 for finding an optimal solution. Figure 17 illustrates the use of this algorithm.

Case 3: Parallel Operations on Input, Identical Machines

In this case we have two parallel operations for each job, where each operation can be put on either machine, and the machines have identical processing characteristics (i.e., each operation will take the same amount of time no matter which machine processes which operation.)

Before we can include the b_i operations in the sequencing problem, we have to solve the parallel machine sequencing problem. Let a_i and a'_i represent the processing times for the two parallel operations for the ith job. We want to assign the a_i's and the a'_i's to machines A and A*, such that we minimize total processing time for the operations. Once we have a minimum, we know which operations for each job should appear on which machines. The sequencing of the jobs will not affect this minimum, since a machine can begin processing one job as soon as it finishes processing the previous job. (This is so because for any job, no operation precedes operations a_i and a'_i.) A lower bound for the sequence J_r is:



Jobset

i	1	2	3	4
a_i	13	7	26	2
b_i	3	12	9	6
b^*_i	12	16	7	1

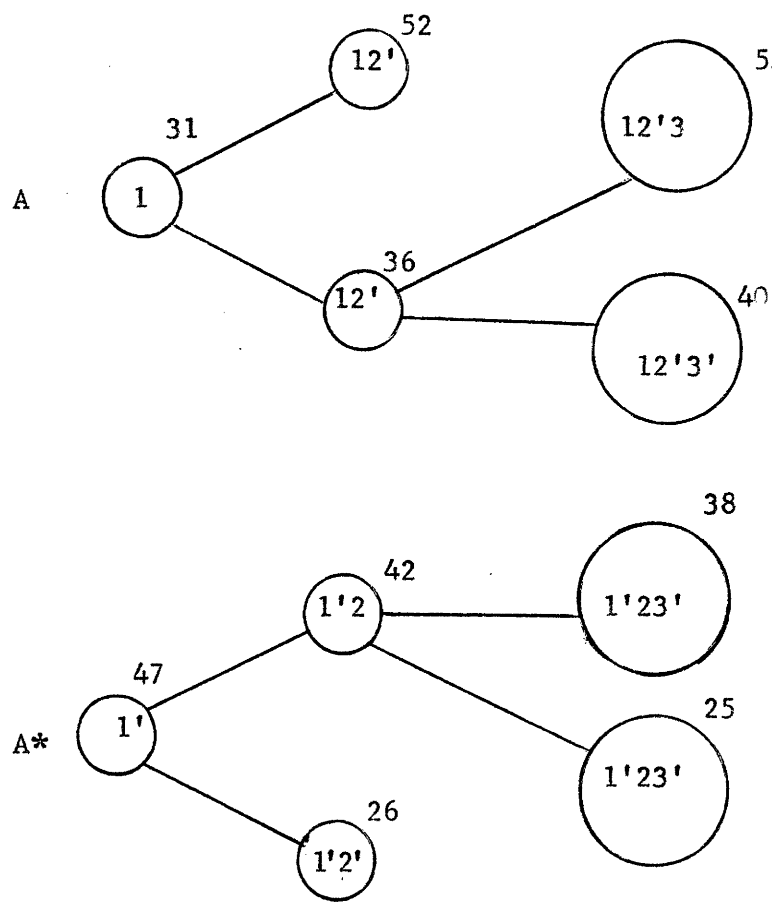
An optimal sequence is 2-1-4-3

Figure 17. Tree structure for 4-job 3-machine problem where 2 different operations are performed on different machines in parallel after the third operation.

$$LB(J_r) = \max \begin{bmatrix} TIMEA(J_r) + \begin{cases} \sum_{\bar{J}_r} \min(a_i, a_i'), & a_r > a_r^* \\ \sum_{\bar{J}_r} \max(a_i, a_i'), & a_r < a_r^* \end{cases} \\ \\ TIMEA^*(J_r) + \begin{cases} \sum_{\bar{J}_r} \min(a_i, a_i'), & a_r^* > a_r \\ \sum_{\bar{J}_r} \max(a_i, a_i'), & a_r^* < a_r \end{cases} \end{bmatrix}$$

Figure 18 illustrates the use of this algorithm. Choose any job to start. In Figure 18, job 1 was chosen as the first job.

1. Compute the lower bounds as described above. Keep a separate tree for each machine.
2. Extend the trees by adding both possible operations to both trees. Compute the lower bounds for the new nodes.
3. Group the new nodes in pairs, each pair representing a feasible schedule (e.g., 12 for machine A, and 1'2' for machine A*; 12' for machine A, and 1'2 for machine A*).
4. Compute the lower bound for each pair of nodes as described above.
5. Keep the minimum of the two lower bounds, and compare it with the previous lower bound.
6. If the previous lower bound is smaller, its nodes define an optimal sequencing for the two machines.
7. If the new lower bound is equal to or less than the previous lower bound, than the new lower bound is used to continue



i	1	2	3	4
a_i	13	7	26	2
a'_i	3	12	9	6
b_i	12	16	7	1

An optimal sequence is 1-2'-3'-4' & 1'-2-3-4

Figure 18. Tree structure for 4-job, 2-machine problem, where 2 identical operations are performed on identical machines in parallel.

branching with step 2, unless the new lower bound has scheduled n-1 jobs, in which case it defines an optimal sequencing for the two machines.

Now the b_i operations can be included in the sequencing problem. Compute a lower bound as described under "Parallel Operations on Input." Follow the algorithm in Chapter 4 for finding an optimal solution. Figure 19 illustrates the use of this algorithm.

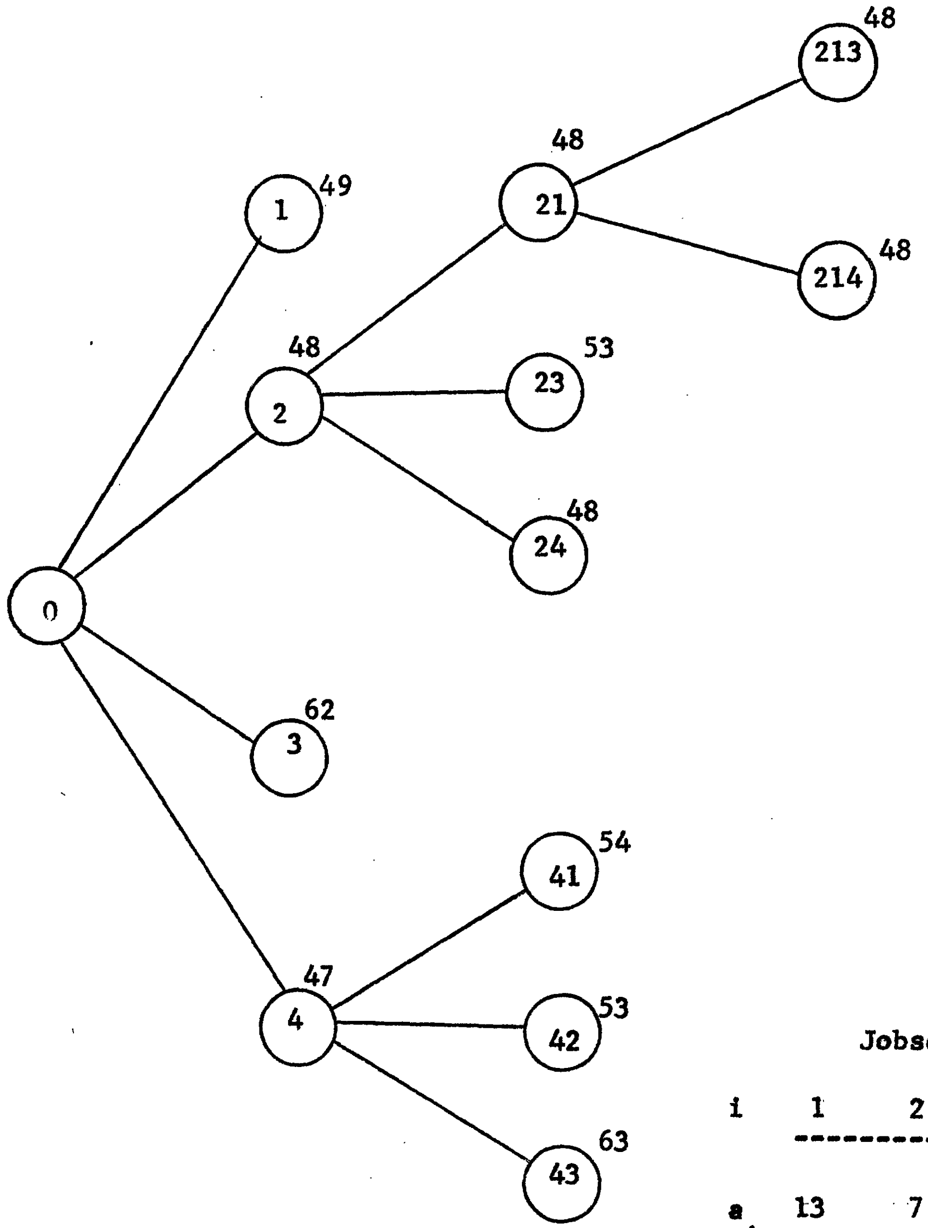
<u>Case 4: Parallel Operations on Output, Identical Machines</u>

As in case 3, we have two parallel operations for each job, where each operation can be put on either machine, and the machines have identical processing characteristics.

We want to use the lower bound formulas listed under "Parallel Operations on Output." To do so, compute $\sum_{J_r} b_i$ and $\sum_{J_r} b_i^*$ in the way that a_i and a_i^* sums were computed for case 3. After computing the lower bounds using the algorithm in case 3 and formula under "Parallel Operations on Output," we can follow the algorithm in Chapter 4 for finding an optimal solution. Figure 20 illustrates case 4.

Case 5: <u>Parallel Operations on Input, Same Operation, Different Machines</u>

In this case we have two parallel operations for each job, where each operation can be put on either machine, but the machines have different operating characteristics (i.e., each operation takes longer to complete on one machine than it does on the other).

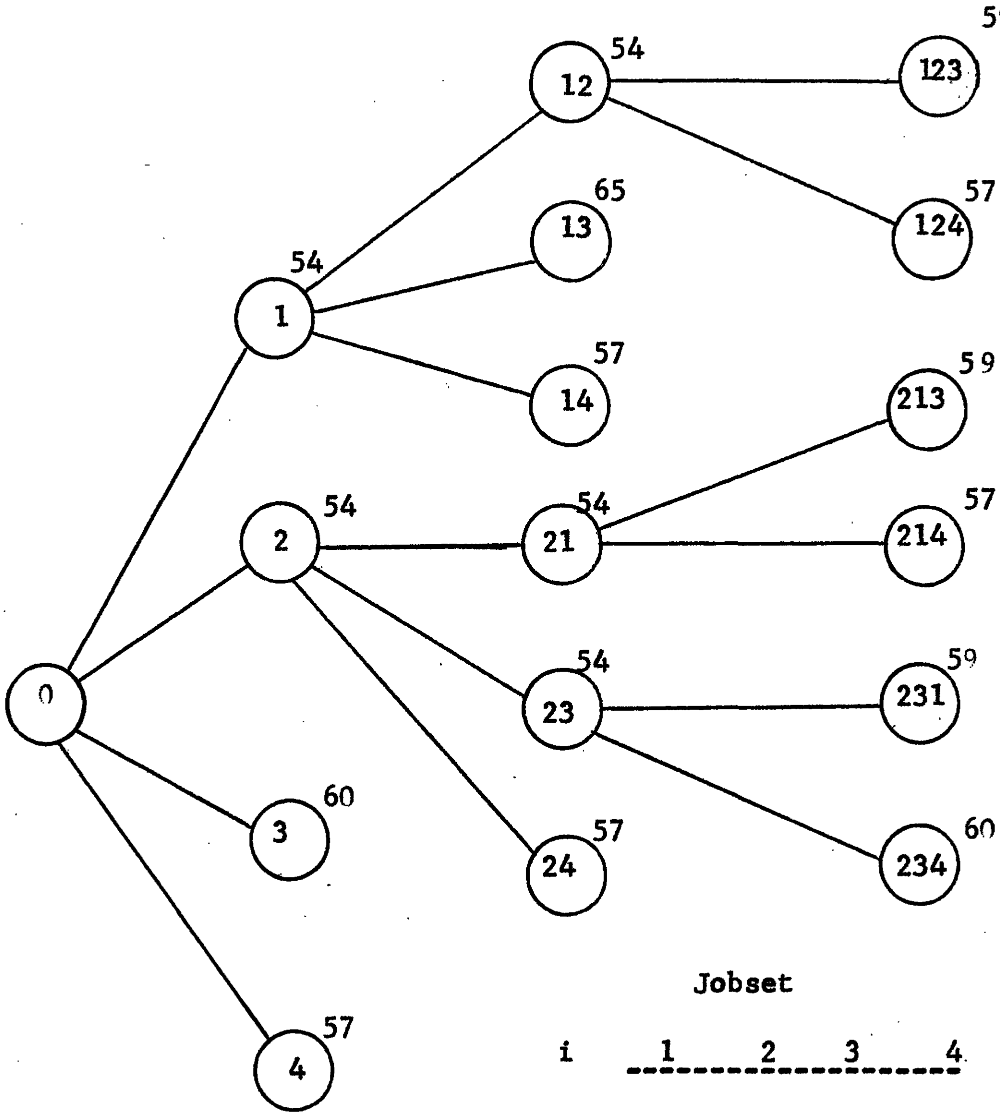


Jobset

i	1	2	3	4
a_i	13	7	26	2
a'_i	3	12	9	6
b_i	12	16	7	1

An optimal sequence is 2-1-3-4

Figure 19. Tree structure for 4-job, 3-machine problem, where 2 identical operations are performed on identical machines in parallel before the third operation.



Jobset

i	1	2	3	4
a_i	13	7	26	2
b_i	3	12	9	6
b'_i	12	16	7	1

An optimal sequence is 2-1'-4-3 & 2'-1-4'-3'

Figure 20. Tree structure for 4-job, 3-machine problem, where 2 identical operations are performed on identical machines in parallel after the third operation.

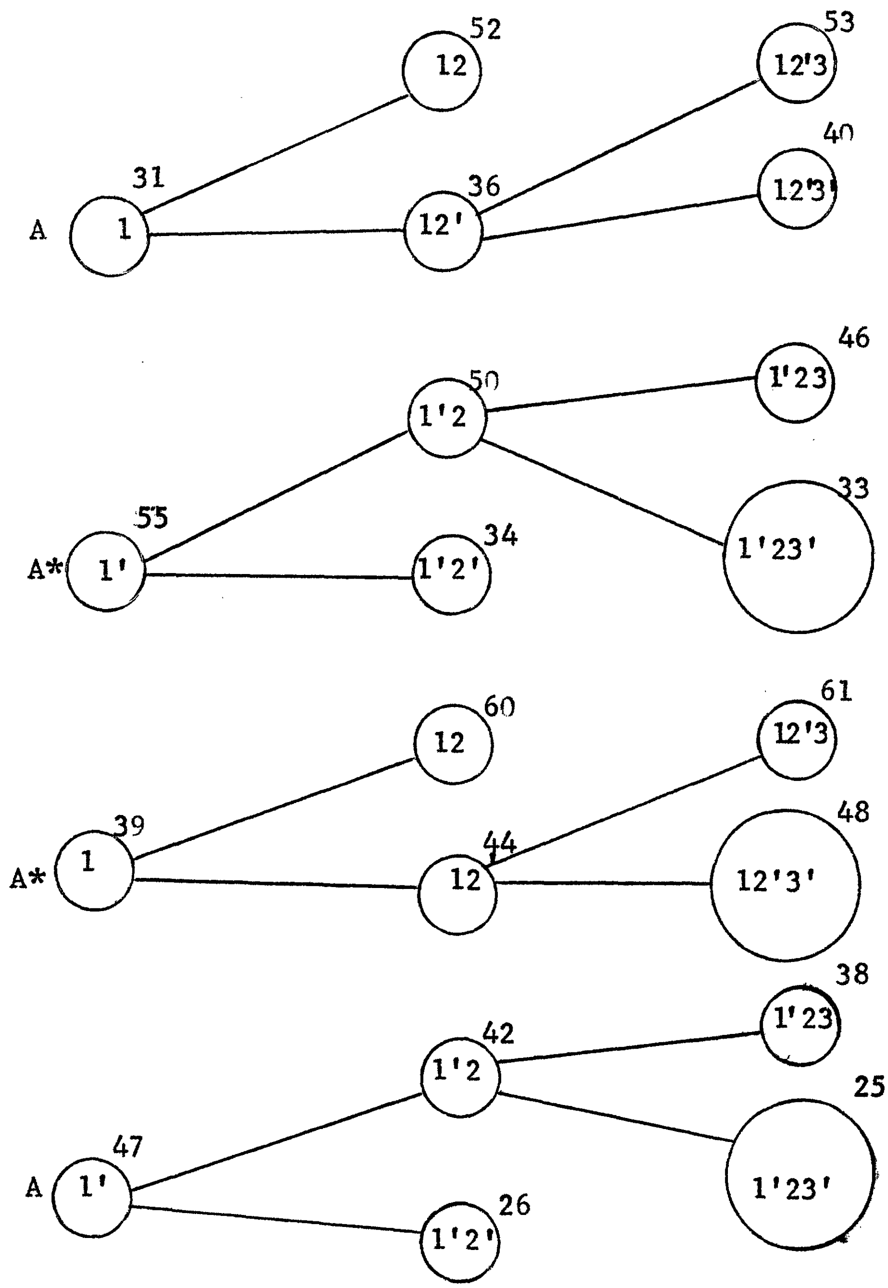
The technique used is similar to that for case 3, except that two pairs of trees are maintained instead of one pair. As in case 3, choose any job to start. As shown in Figure 21, assign one operation to machine A in one tree pair, and to machine A* in the other tree pair. For each tree pair follow the algorithm in case 3 to obtain an optimal sequence for that tree pair. Compare the operation times for the two sequences, and choose the smaller as the optimal sequence.

Now include the b_i operations in the sequencing problem, as shown in Figure 22. Compute a lower bound as described under "Parallel Operations on Input." Follow the algorithm in Chapter 4 for finding an optimal sequence.

Case 6: Parallel Operations on Output, Same Operation Different Machines

As in case 3, we have two parallel operations for each job, where each operation can be put on either machine, but the machines have different operating characteristics.

We want to use the lower bound formulas listed under "Parallel Operations on Output." To do so, compute $\sum_{J_r} b_i$ and $\sum_{J_r} b^*_i$ in the way the a_i and a^*_i sums were computed for case 5. After the lower bounds are computed in this manner, we can follow the algorithm in Chapter 4 for finding an optimal sequence, as shown in Figure 23.

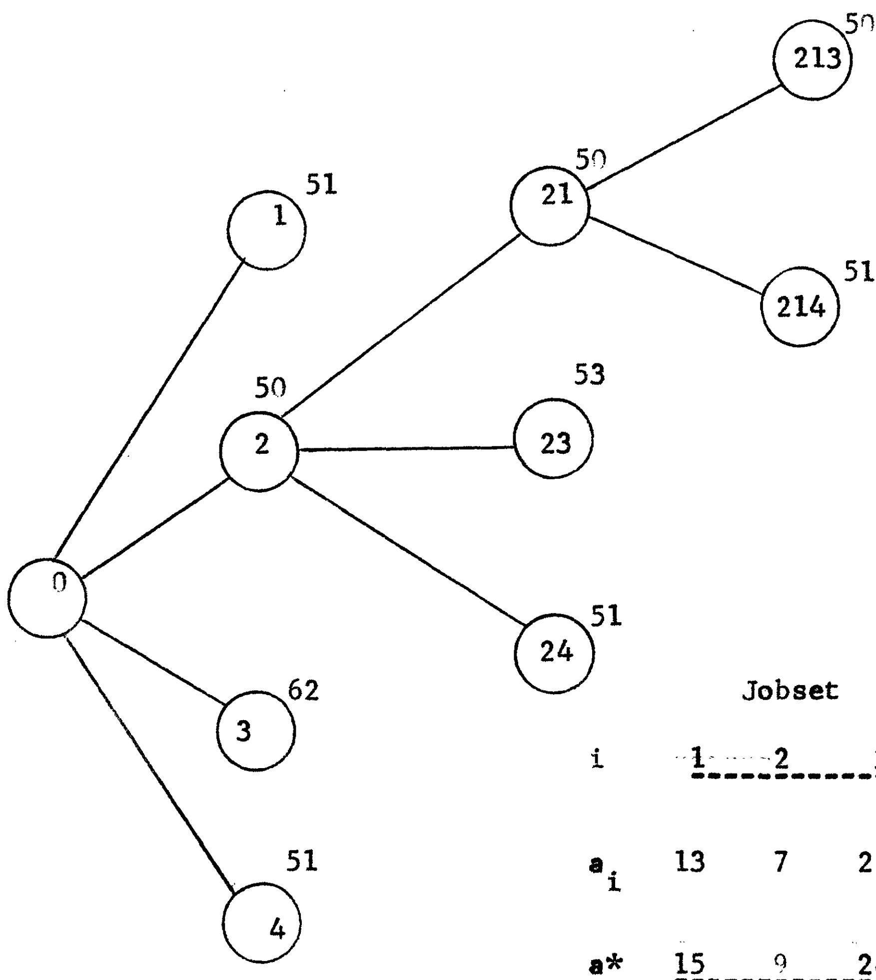


An optimal sequence is 1'-2-3-4' on machine a & 1-2'-3'-4 on machine a*

Jobset

i	1	2	3	4
a_i	13	7	26	2
a^*_i	15	9	28	4
a'_i	3	12	9	6
a'^*_i	5	14	11	8
b_i	12	16	7	1

Figure 21. Tree structure for 4-job, 2-machine problem, where 2 identical operations are performed on different machines in parallel.

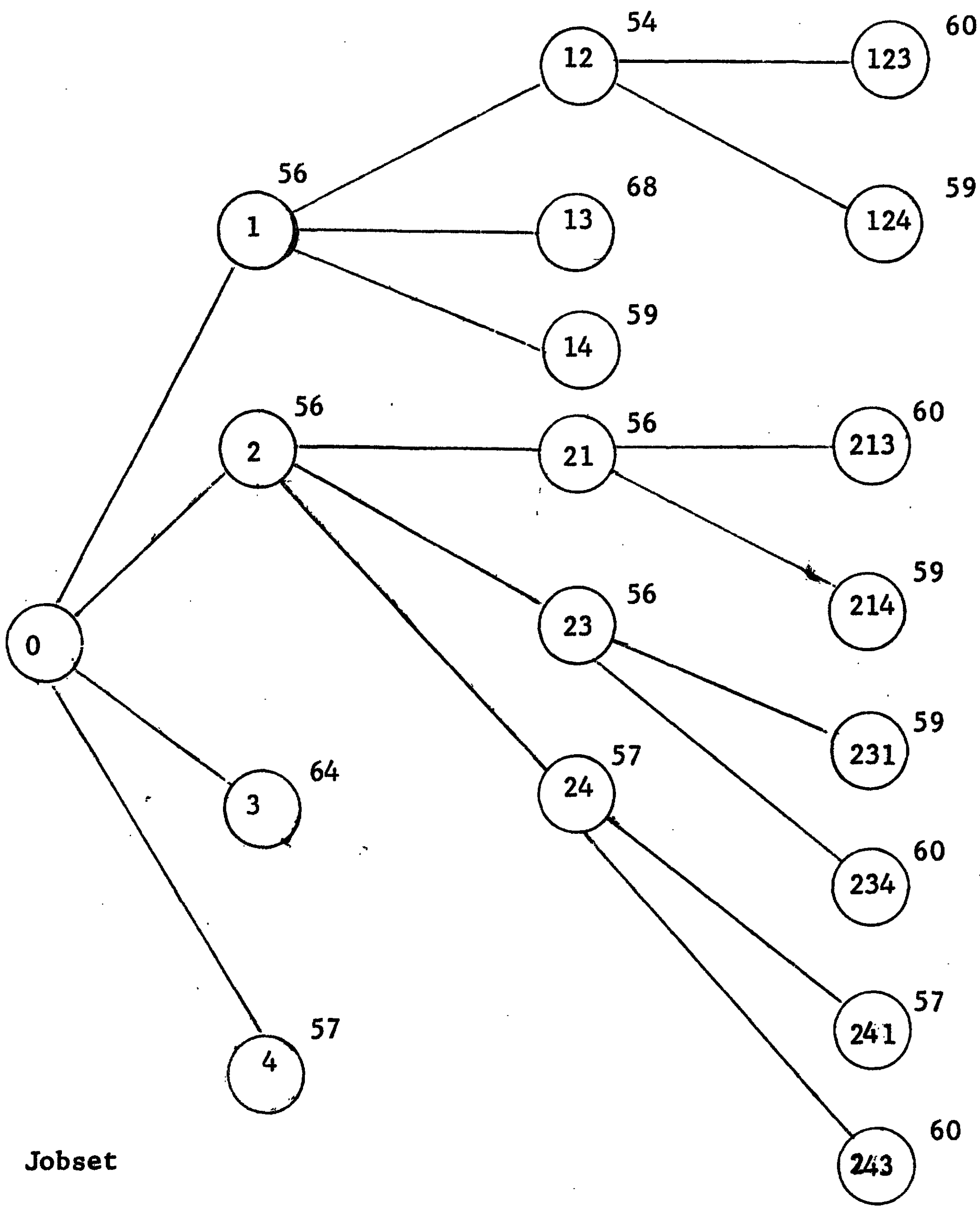


Jobset

i	1	2	3	4
a_i	13	7	26	2
a^*_i	15	9	28	4
a'	3	12	9	6
a'^*_i	5	14	11	8
b_i	12	16	7	1

An optimal sequence is 2-1-3-4

Figure 22. Tree structure for 4-job, 3-machine problem, where 2 identical operations are performed on different machines in parallel before the third operation.



Jobset

i	1	2	3	4
a_i	13	7	26	2
b_i	3	12	9	6
b^*_i	5	14	11	8
b'	12	16	7	1
b'^*_i	14	18	9	3

An optimal sequence is 2' - 4' - 1' - 3 on machine b & 2 - 4 - 1' - 3' on machine b*

Figure 23. Tree structure for 4-job, 3-machine problem, where 2 identical operations are performed on different machines in parallel after the third operation.

Chapter 6

SCHEDULING RULE

We can now combine the tools developed in the preceding three chapters to schedule the programs run on the computer. The schedule period is a week. Most jobs enter the computer center at the beginning of one day and must be completed by the end of that day. Other jobs have different turnaround times up to a maximum of one week, but all jobs must be completed during the week they enter the computer center. The foreground areas are assigned a normal use for the week (e.g., one area for background program input, the other for background program output), but there may be other foreground programs to schedule. Some background programs form a program set which must be executed in sequence with no other programs intervening. Some background programs need devices which one or both of the foreground programs use. Such background programs cannot be multiprogrammed with other programs, if it results in a conflict of this sort.

1. Remove from the list of programs to be scheduled those background programs which cause conflicts, those foreground programs which are not assigned to a foreground area as the normal program for that area, and those programs whose entrance and completion times are not the beginning and end of the same day.
2. Combine background program sets into a single conceptual program by summing the processing times and foreground program times across programs.

3. For the remaining programs (the normal foreground programs; background programs which do not cause conflicts and whose entrance and completion times are the beginning and end of the same day) compute optimal daily schedules as follows:
   a. Compute the individual program times using the simulation method in Chapter 3.
   b. Sequence the programs in each day using the appropriate algorithm in Chapter 4 or 5.
4. For each day's schedule, starting with the first day, and proceeding day by day until the end of the week, do the following:
   a. Adjust the part of the schedule which includes the background program sets. This is required, since we can now overlap one program's processing with other programs' input or output transcription operations.
   b. Insert the daily programs which have been removed from the list because of device conflicts, foreground area usage, or entrance and completion times.
   c. Insert as many programs as possible that have longer than one day turnaround times, starting with those with the nearest due date.

This scheduling rule is in the form of a computer program. Appendix A describes the program's input and output formats. Appendix B contains the program flow charts and FORTRAN coding. Appendix C contains examples of program input and output.

Chapter 7

CONCLUSIONS AND SUGGESTIONS FOR FUTURE STUDY

A scheduling program has been developed which provides good or "near" optimal schedules for a set of programs run on a multiprogrammed computer. The program provides more efficient schedules for the computer than could be provided by a computer operator without using the program.

The scheduling rule developed here consists of a simulation method which is used to calculate the running times for programs in the multiprogramming environment, and sequencing algorithms which are used to schedule the programs after the running times have been calculated. New branch and bound algorithms have been developed for the n-job, 3-machine sequencing problem, where 2 of the machines can operate on the same job in parallel.

Future research in computer scheduling should be directed towards more complex environments than the one studied here, and towards providing accurate data for use with the scheduling program.

The environment studied here is a static one. That is, all the programs to be scheduled during a period are known before the schedule is made. Within the context of the IBM System/360 Disk Operating System, research should be directed towards developing a dynamic scheduling rule. Such a scheduling rule would consist of a computer program which is always in the computer's main storage, ready to be executed whenever a new program enters the computer center. This scheduling program would relieve the operator of the burden of producing a new schedule whenever the scheduling environment changed. Instead, the scheduling program would keep track of the then current schedule, dispatch programs when an area of main storage is free, and modify the schedule when there are changes in the environment.

The ability of the scheduling program to produce good schedules is dependent upon the accuracy of the input data which describes program characteristics. Before more complex scheduling rules are produced, research should be directed towards developing accurate data on program characteristics.

Appendix A

PROGRAM USER MANUAL

This appendix contains the information needed to use the scheduling program. The scheduling program is an efficient means for producing a weekly schedule for programs run on the IBM System/360 under control of the IBM System/360 Disk Operating System (DOS/360). DOS/360 provides a multiprogramming environment where main storage is divided into three areas (a background area and two foreground areas). Input to the scheduling program consists of statements describing the computer, the installation (including normal foreground area assignments), and the programs to be scheduled. Output from the program consists of a suggested schedule, showing program sequence and clock time for the scheduled week.

The programs to be scheduled are divided into three categories: weekly programs, partial working day programs, and full working day programs. Each program category may consist of background programs, foreground programs, or both background and foreground programs. Weekly programs are ready for processing on one day of the week, but do not have to be completed until some day later in the week. Partial working day programs must be completed the same day they are ready for processing, but in a shorter time span than full working day programs. Full working day programs are ready for processing at the start of a day and must be completed by the end of the same day.

The scheduling program checks all input statements for proper syntax. If there are any errors in a statement, the program prints an error message, prints the statement in error, and does not produce a schedule. The statements in error can then be corrected and the scheduling program can be rerun.

Input Formats

The input to the scheduling program consists of a dictionary statement, hardware statements, installation statements, and job statements.

Dictionary Statement: The dictionary statement is used to define the syntactic characters used by the scheduling program. It must be the first statement in the input stream. The format of this statement is:

Columns	Contents
1-24	ABCDFHIJOPRSTUXZ01234567
25-36	Must be blank
37-80	Should be blank

Hardware Statements: The hardware statements describe the physical characteristics of the computer. They consist of an HC statement, one or more HD statements, and an HZ statement.

The HC statement describes the physical characteristics of the central processing unit. This statement must immediately follow the dictionary statement. If the HC statement is in error, the run is terminated immediately, since the error may have been caused by the absence of the dictionary statement. The format of the HC statement is:

Column(s)	Contents
1	H
2	C
3-6	Central processing unit model number (e.g., 2030)
7-9	Average instruction time in tenths of a microsecond
10-12	Interference time per byte on selector channels in hundredths of a microsecond
13-15	Interference time per byte on multiplexor channel in tenths of a microsecond. This time may be overridden for individual devices by use of the HD statement.
16-80	Should be blank

The HD statements describe the physical characteristics of the input/output devices. These statements must immediately follow the HC statement. There must be at least one HD statement. The maximum number of HD statements is determined when the scheduling program is compiled. Initially, the maximum number is ten. The format of these statements is:

Column(s)	Contents
1	H
2	D
3-6	Device type (e.g., 2311)
7-8	Model number; used to distinguish devices with the same device type, but with different physical characteristics. The field must be used to distinguish (1) the punch and read units of a card read punch, (2) 7-track and 9-track tape drives, and (3) devices with the same device type that have different speeds.
9	Device class--D for disk, T for tape, U for unit record
10-16 (D or T in 9)	Device speed in bytes per second
10-12 (U in 9)	Record size in bytes

13-16 (U in 9)	Device speed in records per minute. This speed may be overridden for individual files by use of JBIO and JFIO statements.
17-18	Device set-up time in tenths of a minute
19-21 (D in 9)	Average latency time for device in milliseconds. This time may be overridden for individual files by use of the IF, JBIO, and JFIO statements.
19-21 (U in 9)	Start-stop time for device in tenths of a millisecond.
19-21 (T in 9)	Should be blank
22-24 (D in 9)	Average seek time for device in milliseconds. This time may be overridden for individual files by use of the IF, JBIO, and JFIO statements.
22-24 (T in 9)	Rewind time for device in tenths of a minute
22-24 (U in 9)	Should be blank
25-27 (T in 9)	Device density in bytes per inch
25-27 (D or U in 9)	Should be blank
28-30	Interference time per byte on multiplexor channel in tenths of a microsecond, if desired to override time on HC statement. Otherwise should be blank.
31-80	Should be blank

The HZ statement indicates the end of the hardware statements. It must immediately follow the last HD statement. The format of this statement is:

Column	Contents
1	H
2	Z

<u>Installation Statements</u>: The installation statements describe the input/output and main storage configurations, and the work week. They consist of one or more ID statements, two IF statements, an IS statement, and an IZ statement.

The ID statements relate input/output unit numbers to device descriptions. These statements must immediately follow the JZ statement. There must be at least one ID statement. The maximum number of unit numbers is determined when the scheduling program is compiled. Initially the maximum number is 20. The format of these statements is:

Column(s)	Contents
1	I
2	D
3-6	Device type
7-8	Model number. The device type and model number must be the same as a device type and model number that appears on a valid HD statement.
9-10	Number of unit numbers in this statement, an integer from 1 to 23
11-79	Unit numbers in hexadecimal, each unit number taking up three columns
80	Any non-blank character, if there are more ID statements, Otherwise, blank.

The IF statements describe the normal usage of the foreground areas. These statements must immediately follow the last ID statement. There must be two IF statements, one for each foreground area. The format of these statements is:

Column(s)	Contents
1	I
2	F
3	1 for the first statement (foreground area one) and 2 for the second statement (foreground area two).
4-5	R for card reader transcription program, P for printer transcription program, TI for telecommunications input transcription program, TO for telecommunications output transcription program, or blank if no program assigned to this area.

6-13 (4-5 non-blank)	Program name
14-16 (4-5 non-blank)	Unit number for peripheral operation. The device class of this unit number must be U.
17-19 (4-5 non-blank)	Unit number for intermediate storage. The device class of this unit number must be D or T.
20-22 (4-5 non-blank)	Number of instructions executed per record.
23-24 (4-5 non-blank)	Blocking factor on intermediate storage device.
25-27 (4-5 non-blank)	Average latency time for intermediate storage device (device class D), if desired to override time on HD statement. Otherwise, blank.
28-30 (4-5 non-blank)	Average seek time for intermediate storage device (device class D), if desired to override time on HD statement. Otherwise, blank.
6-30 (4-5 blank)	Should be blank
31-80	Should be blank

The IS statement describes the work week in terms of the time the installation is active during each day. This statement must immediately follow the second IF statement. The format of this statement is:

Column(s)	Contents
1	I
2	S
3	Number of days in week, from 1 to 7
4-59	Begin time and end time for each day in the form HHMMHHMM, where HH is an hour from 00 to 23 and MM is a minute from 00 to 59.
60-80	Should be blank

The IZ statement indicates the end of the installation statements. It must immediately follow the IS statement. The format of this statement is:

Column	Contents
1	I
2	Z
3-80	Should be blank

Job Statements: The job statements describe the programs to be scheduled. Each background program description consists of a JB statement, one or more JBD statements, two JBF statements, and one or more JBIO statements. Each foreground program description consists of a JF statement, one or more JFD statements and one or more JFIO statements. A JZ statement follows the last program description.

The descriptions for weekly programs must precede the descriptions for daily programs. The daily program descriptions must be ordered by day. Within each day, the partial working day program descriptions must precede the full working day program descriptions. Within each program category (weekly, partial working day for each day, and full working day for each day), background program descriptions must precede foreground program descriptions. If a background job consists of two or more job steps, the program descriptions for the steps must appear in sequence with no other program descriptions intervening, and the job name and begin- and end-times for each step must be the same.

The maximum number of programs in each category is determined when the scheduling program is compiled. Initially, the maximum number of weekly programs is 10, the maximum number of partial working day programs is 5 for each day, and the maximum number of full working day programs is 5 for each day.

The JB and JF statements contain the names of the programs to be scheduled and describe the scheduling constraints for each program. The first of these statements must immediately follow the IZ statement. Additional statements of this type must immediately follow the last JBIO and JFIO statement for the previous program. The format of these statements is:

Column(s)	Contents
1	J
2	Blank for weekly programs or an integer from 1 to 7 corresponding to a day on the IS statement for daily programs.
3	B for background programs or F for foreground programs
4-5	Should be blank
6-13 (B in 3)	Job name
6-13 (F in 3)	Program name
14-21 (B in 3)	Step name
14-21 (F in 3)	Should be blank
22-26	Time when program is ready for processing, in the form DHHMM, where D is a day from 1 to 7, HH is an hour from 00 to 23, and MM is a minute from 00 to 59.
27-31	Time by which program must be completed, in the form DHHMM, where D is a day from 1 to 7, HH is an hour from 00 to 23, and MM is a minute from 00 to 59. For weekly programs, column 22 does not equal column 27. For daily programs those columns are the same. For full working day programs, columns 23-26 and 28-31 must be the same as the begin-time and end-time for day D. For partial working day programs at least one of those times must be different than the corresponding times on the IS statement.
32-36	Fixed number of instructions executed by program regardless of the volumes of its files.

37-41	Number of instructions executed per record for files indicated on JBIO or JFIO statements
42-49 (B in 3)	For jobs which consist of two or more steps, the step name of the preceding program description. Otherwise, should be blank.
42-49 (F in 3)	Should be blank
50-51	The percentage of machine instructions which cannot be overlapped with input/output operations
52-80	Should be blank

The JBD and JFD statements relate file names to unit numbers. These statements must immediately follow the JB and JF statements. There must be at least one of these statements for each program description. The maximum number of files for each program is determined when the scheduling program is compiled. Initially this number is 20. The format of these statements is: .

Column(s)	Contents
1	J
2	Same as first statement for this program description
3	Same as first statement for this program description
4	D
5	Should be blank
6	Number of file names on this statement, an integer from 1 to 7
7-76	One or more relations between a file name and a unit, each consisting of a 7 character file name and a 3 hexadecimal digit unit number. The unit number must be the same as one that appears on a valid ID statement.
77-79	Should be blank
80	Any non-blank character, if there are more of these statements for this program description. Otherwise, blank.

The JBF statements define the unit record files which can be processed by the foreground transcription programs described by the IF statements. These statements must immediately follow the last JBD statement for each background program description. There must be two JBF statements for each background program description. The format of the statements is:

Column(s)	Contents
1	J
2	Same as first statement for this program description
3	B
4	F
5	Should be blank
6-7	R if file can be processed by card reader transcription program, P if file can be processed by printer transcription program, T if file can be processed by telecommunications input transcription program, TO if file can be processed by telecommunications output transcription program, or blank if there is no file name on this statement.
8-14 (6-7 non-blank)	File name. Must be the same as a file name that appears on a valid JBD statement for this program description. The device class of the unit number for this file must be U.
8-14 (6-7 blank)	Should be blank
15-80	Should be blank

The JBIO and JFIO statements describe the characteristics of each file processed by the program. The JBIO statements must immediately follow the second JBF statement for each program description. The JFIO statements must immediately follow the last JFD statement for each program. There must be one JBIO statement for each file name that appears on a valid JBD statement. Similarly, there must be one JFIO statement for each file name that appears on a valid JFD statement.

The format of these statements is:

Column(s)	Contents
1	J
2	Same as first statement for this program description
3	Same as first statement for this program description
4	I
5	0
6-12	File name
13-20	Number of records in file
21-25	Record size in bytes. If the device class of the unit assigned to this file is U, then this field is ignored and the device's record size is used.
26-27	Blocking factor. If the device class of the unit assigned to this file is U, then this field is ignored, and the blocking factor is set equal to 1.
28	X if the number of instructions executed by this program is a function of this file's size. Otherwise, blank. If X, then the number of records in this file is multiplied by the number in columns 37-41 of the JB or JF statement, and the product is added to the fixed number of instructions.
29	A character to indicate valid device classes for the unit assigned to this file. A if device class can be D, T, U; D if device class can be D; H if device class can be D or T; T if device class can be T; or U if device class can be U.
30-32	Average latency time for device (device class D), if desired to override time on HD statement. Otherwise, blank.
33-35	Average seek time for device (device class D), if desired to override time on HD statement. Otherwise, blank.

36-39	Factor in thousandths (device class U), if desired to modify device speed on HD statement. Otherwise, blank. If a number appears here, the device speed is multiplied by it.
40-79	Should be blank
80	Any non-blank character, if more statements of this type. Otherwise, blank.

The JZ statement indicates the end of the job statements. It must immediately follow the last statement for the last program description. The format of this statement is:

Column	Contents
1	J
2	Z

Output Formats

Output from the scheduling program consists of a summary of program restrictions, a summary of the input statements, error messages, and the schedules for each main storage area for each day.

Program Restriction Summary: The first page of output contains the program maximums for number of device types, unit numbers, files, full working day programs per day, partial working day programs per day, and weekly programs.

Input Summary: Summaries are printed for the central processing unit, each device type and model number, both foreground area assignments, the weekly schedule, and each program.

The central processing unit summary consists of a line for each of the fields on the HC statement.

Each device and model number summary consists of a line for each field on the HD statement followed by the unit numbers related to the device type and model number via the ID statements.

The foreground area summaries consist of a line for each field on the IF statements.

The weekly schedule summary contains the begin and end times for each day in the week.

Each program summary consists of three parts. The first part contains a line for each field on the JB or JF statement. The second part contains a line for each file assigned to a unit number as shown on the JBD or JFD statement, except for those background program files that can be processed by the foreground transcription programs. If the file cannot be processed by the foreground program because of device conflicts, the unit number printed is the one that appeared on the JBD statement. If the file can be processed by only one foreground program, the unit number printed is the intermediate storage device used by the foreground program. If the file can be processed by both foreground programs, no unit number is printed here. Instead the unit number assigned is printed just before the schedule is printed. The third part of the program summary contains a header line for each file, followed by a line for each field on the JBIO or JFIO statements.

Error Messages: Error messages consist of a heading, followed by a message, (usually) followed by the input statement in error, as shown in the following example. Some errors do not refer to a specific statement and, therefore, no input statement follows the message.

```
**********ERROR**********
UNIT NUMBER 28I NOT DEFINED BY AN ID STATEMENT
J1BD 1 MASTER 28I
```

Schedules: A separate schedule is printed for each main storage area for each day. These schedules are followed by a list of programs that could not be included in the schedule due to time constraints. The format of the background area schedule is:

Columns	Contents
1-2	Hour on
3	Blank
4-5	Minute on
6	Blank
7-8	Hour off
9	Blank
10-11	Minute off
12	Blank
13-20	Job name
21	Blank
22-29	Step name
30	Blank
31-32	Set-up time in minutes
33	Blank
.34	Rewind time in minutes
35	Blank
36-39	Central processing unit work time, in the form HHMM, where HH is hours and MM is minutes
40	Blank
41-44	Central processing unit wait time, in the form HHMM, where HH is hours MM is minutes
45	Blank
46-48	Unit address in hexadecimal
49	Blank
50-53	Unit work time, in the form HHMM, where HH is hours and MM is minutes
54	Blank
55-58	Unit wait time, in the form HHMM, where HH is hours and MM is minutes
59	Blank
60-128	Additional unit addresses, work times, and wait times. If additional lines are needed, they are printed starting with Column 46.

The format of each foreground area schedule is:

Column(s)	Contents
1-2	Hour on
3	Blank
4-5	Minute on
6	Blank
7-8	Hour off
9	Blank
10-11	Minute off
12	Blank
13-20	Program name
21	Blank
22-29	Job name, if program is a transcription program. Otherwise, blank.
30	Blank
31-38	Step name, if program is a transcription program. Otherwise, blank.
39	Blank
40-41	Set-up time in minutes
42	Blank
43	Rewind time in minutes
44	Blank
45-48	Central processing unit work time, in the form HHMM, where HH is hours and MM is minutes
49	Blank
50-53	Central processing unit wait time, in the form HHMM, where HH is hours and MM is minutes
54	Blank
55-57	Unit address in hexadecimal
58	Blank
59-62	Unit work time, in the form HHMM, where HH is hours and MM is minutes
63	Blank
64-67	Unit wait time, in the form HHMM, where HH is hours and MM is minutes
68	Blank
69-123	Additional unit addresses, work times, and wait times. If additional lines are needed, they are printed starting with Column 55.

Appendix B

PROGRAM LOGIC MANUAL

This appendix is a guide to the logic of the scheduling program. It is intended for persons who want to modify the program or diagnose program malfunctions. Included are descriptions of the program's major components and their storage requirements, the program flow charts, and the FORTRAN source statements.

Program Organization

The scheduling program consists of a main program and eighteen subprograms. Figure 24 shows how these components are related. In Figure 24, vertical lines connect calling programs with the programs they call. For example, MAIN (the main program) calls subprogram IGNSCH which calls subprograms PARAL1, PARAL2, and ERRPRT.

The following text briefly describes the function of the main program and each subprogram and lists the subprograms that each calls.

Main Program: Reads and checks syntax of all input statements except the JBD, JFD, JBIO, and JFIO statements. Writes all program output except those error messages written by subprograms. Adjusts schedules for inclusion of programs which are not scheduled by use of the algorithms.

Subprograms Called: UNITX, DISKX, TAPEX, FRCTN, TIME23, FILE, FILEIO, TOD, SWITCH, JOHNSN, IGNSCH, ERRPRT.

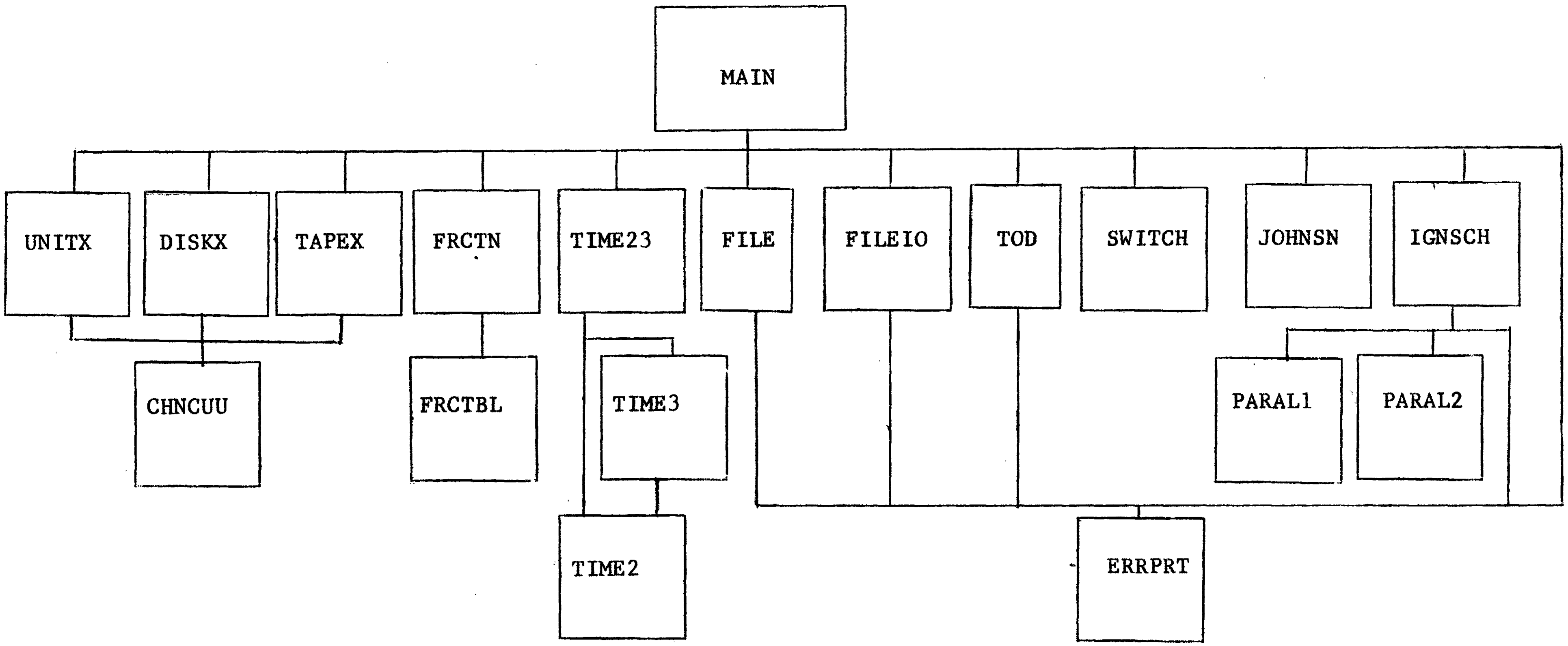


Figure 24. Overall program organization.

<u>Subprogram</u> <u>UNITX</u>: Computes device, channel, CPU, and set up times for unit record devices.

<u>Subprogram</u> <u>Called</u>: CHNCUU

<u>Subprogram</u> <u>DISKX</u>: Computes device, channel, CPU, and set up times for disk devices.

<u>Subprogram</u> <u>Called</u>: CHNCUU

<u>Subprogram</u> <u>TAPEX</u>: Computes device, channel, CPU, set up and rewind times for tape devices.

<u>Subprogram</u> <u>Called</u>: CHNCUU

<u>Subprogram</u> <u>CHNCUU</u>: Determines channel and interference rate for a device.

<u>Subprograms</u> <u>Called</u>: None

<u>Subprogram</u> <u>FRCTN</u>: Separates resource uses into multiplexor mode, burst mode, and CPU portions.

<u>Subprogram</u> <u>Called</u>: FRCTBL

<u>Subprogram</u> <u>FRCTBL</u>: Completes computation of resource uses and then sorts resource uses.

<u>Subprogram</u> <u>Called</u>: None

<u>Subprogram</u> <u>TIME2</u>: Calculates multiprogramming time for second priority program.

<u>Subprograms</u> <u>Called</u>: None

Subprogram TIME3: Calculates multiprogramming time for third priority program.

Subprograms Called: None

Subprogram FILE: Reads and checks syntax of JBD and JFD statements.

Subprogram Called: ERRPRT

Subprogram FILEIO: Reads and checks syntax of JBIO and JFIO statements.

Subprogram Called: ERRPRT

Subprogram TOD: Checks validity of begin and end times on JB and JF statements.

Subprogram Called: ERRPRT

Subprogram SWITCH: Moves program description from one storage area to another storage area.

Subprogram Called: None

Subprogram JOHNSN: Solves two-machine two-operation sequencing problem.

Subprograms Called: None

Subprogram IGNSCH: Solves three-machine three-operation sequencing problem.

Subprograms Called: PARAL1, PARAL2, ERRPRT

Subprogram PARAL1: Solves parallel operation sequencing problem for same operations and same machines.

Subprograms Called: None

Subprogram PARAL2: Solves parallel operation sequencing problem for same operations and different machines.

Subprograms Called: None

Subprogram TIME23: Calculates multiprogramming times for second priority and third priority programs.

Subprograms Called: TIME2, TIME3

Subprogram ERRPRT: Prints error heading and increments error count

Subprograms Called: None

Storage Requirements

The main storage required for the main program and for several subprograms depends upon the values assigned to certain variables in the main program. The variables are:

MDEV, the maximum number of device types and model numbers in the installation.

MCUU, the maximum number of unit numbers in the installation

MFILE, the maximum number of file names per program description

MJOB, the total number of weekly program descriptions (MWJOB), partial working day program descriptions per day (MDJOB - MWJOB), and full working day program descriptions per day (MJOB - MDJOB).

Values are assigned to these variables when the scheduling program is compiled. The following text lists the names of those arrays whose dimensions should be changed when the values assigned to the above variables are changed. In addition, the main storage for subprogram IGNSCH depends upon the value assigned to the variable NLIST, which is defined in subprogram IGNSCH.

Main Program:

MDEV	MCUU	MFILE	MJOB		MJOB-MDJOB	MJOB & MCUU
DEVTYP	CUUPNT	FILPNT	DON	TSEL6A	QL	TDEV
DEVMOD	CUU	FILNAM	HON	TSEL6W	QN	SDEV
DEVCLS			DOFF	TMPXMA	QA	
DEVSPD			HOFF	TMPXMW	QB	
DEVROD			MON	TMPXBA	QC	
DEVSTP			MOFF	TMPXBW	QDX	
DEVLSS			TOTAL	TSETUP	QDP	
DEVSRW			TCPU	TWNDUP	QDPX	
DEVDEN			TNOVLP	JOBNAM	QD	
DEVINT			FLFGPT	STEPNM		
			FLNMPT	FILEFG		
			TSEL1A	FRACM		
			TSEL1W	FRACB		
			TSEL2A	FRACC		
			TSEL2W	FRACMX		
			TSEL3A	FRACBX		
			TSEL3W	FRACCX		
			TSEL4A	ISCHED		
			TSEL4W	SCHED		
			TSEL5A			
			TSEL5W			

UNITX: None

DISKX: None

TAPEX: None

CHNCUU: None

FRCTN: None

FRCTBL: None

TIME2: None

TIME3: None

FILE:

MDEV	MCUU	MFILE
DEVCLS	CUUPNT	FILPNT
	CUU	FILNAM
		A

FILEIO:

MDEV	MCUU	MFILE
DEVCLS	CUUPNT	FILPNT
DEVLSS		FILNAM
DEVSRW		

TOD: None

SWITCH: None

JOHNSN:

MJOB-MDJOB
A
B
N

IGNSCH: The main storage for this subprogram depends upon the value assigned to NLIST, which is the number of nodes created. If N is the number of full working day jobs, then NLIST must be between (.5*N* (N-1)) + 1 and N! The maximum number of full working day jobs that can be scheduled is 9, except for case 4 and case 6, where the maximum is 8.

MJOB - MDJOB		NLIST
A	M	BOUND
B	PA	LIST
C	PB	TIMEA
D	PD	TIMEB
DX	PDX	TIMEC
DP	PDP	POPER
DPX	PDPX	
L	PL	
N		

PARAL1:

MDJOB-MDJOB
A
B
AX
BX
L
LMAX
LMIN

PARAL2:

MDJOB-MDJOB
A
B
C
D
AX
BX
L
L1
LMAX
LMIN

TIME23: None

ERRPT: None

Flowcharts

The following charts can be used to locate the FORTRAN source statements that perform specific program functions. Figure 25 identifies the flowchart symbols. To locate the statements that perform a specific function use the flowcharts to find the flowchart symbol or symbols that describe the function. The FORTRAN statement number of the first source statement used to perform a function appears above the upper left hand corner of the symbol that defines the function.

Flowcharts for the main program have two-digit identifiers. The highest level charts have zero as their first digit. Flowcharts for the subprograms have two-letter identifiers. If there is more than one chart for a subprogram, all the charts have the same first letter. This letter corresponds to the number which appears in columns 74-75 of the source statements for the subprogram.

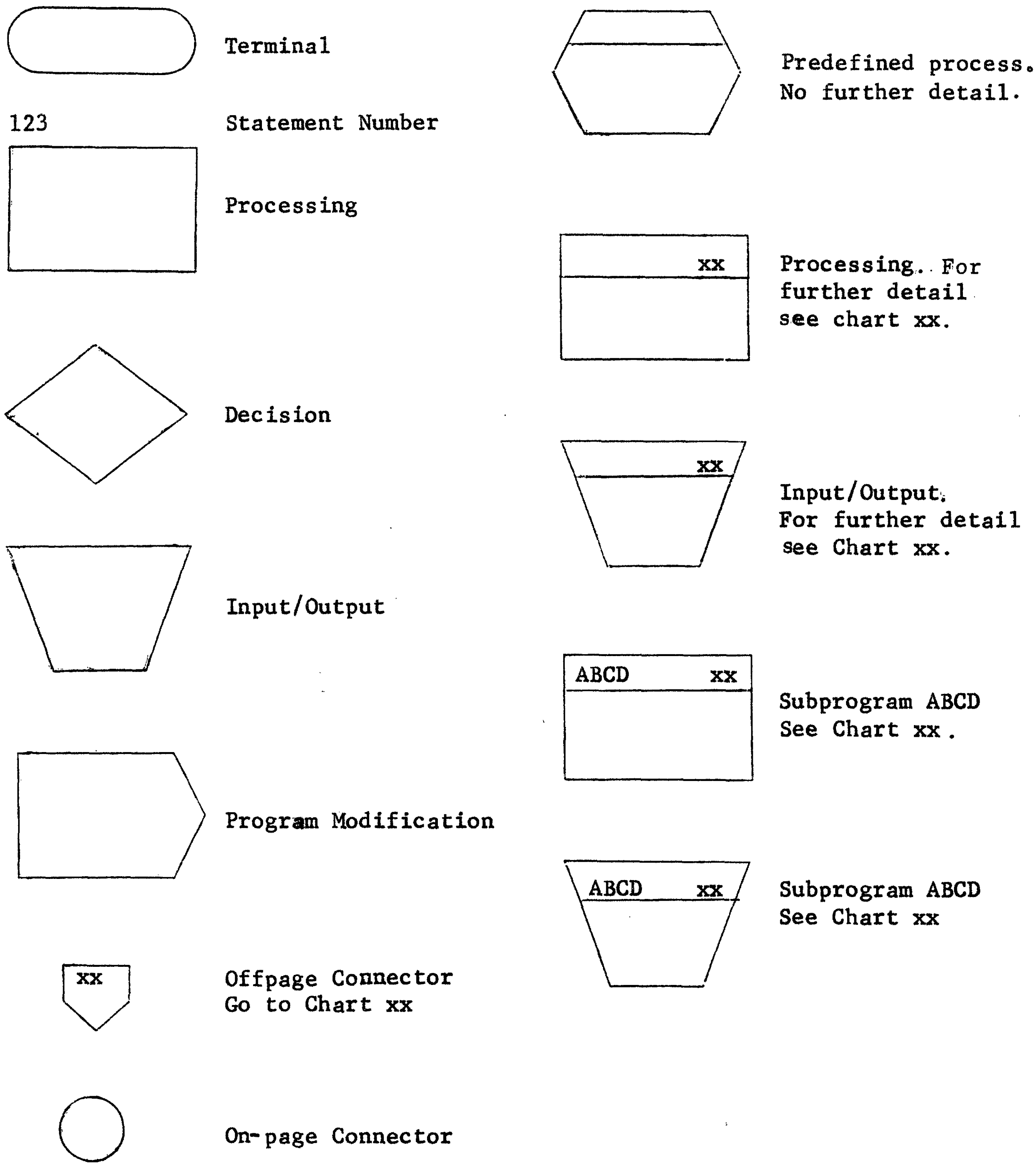
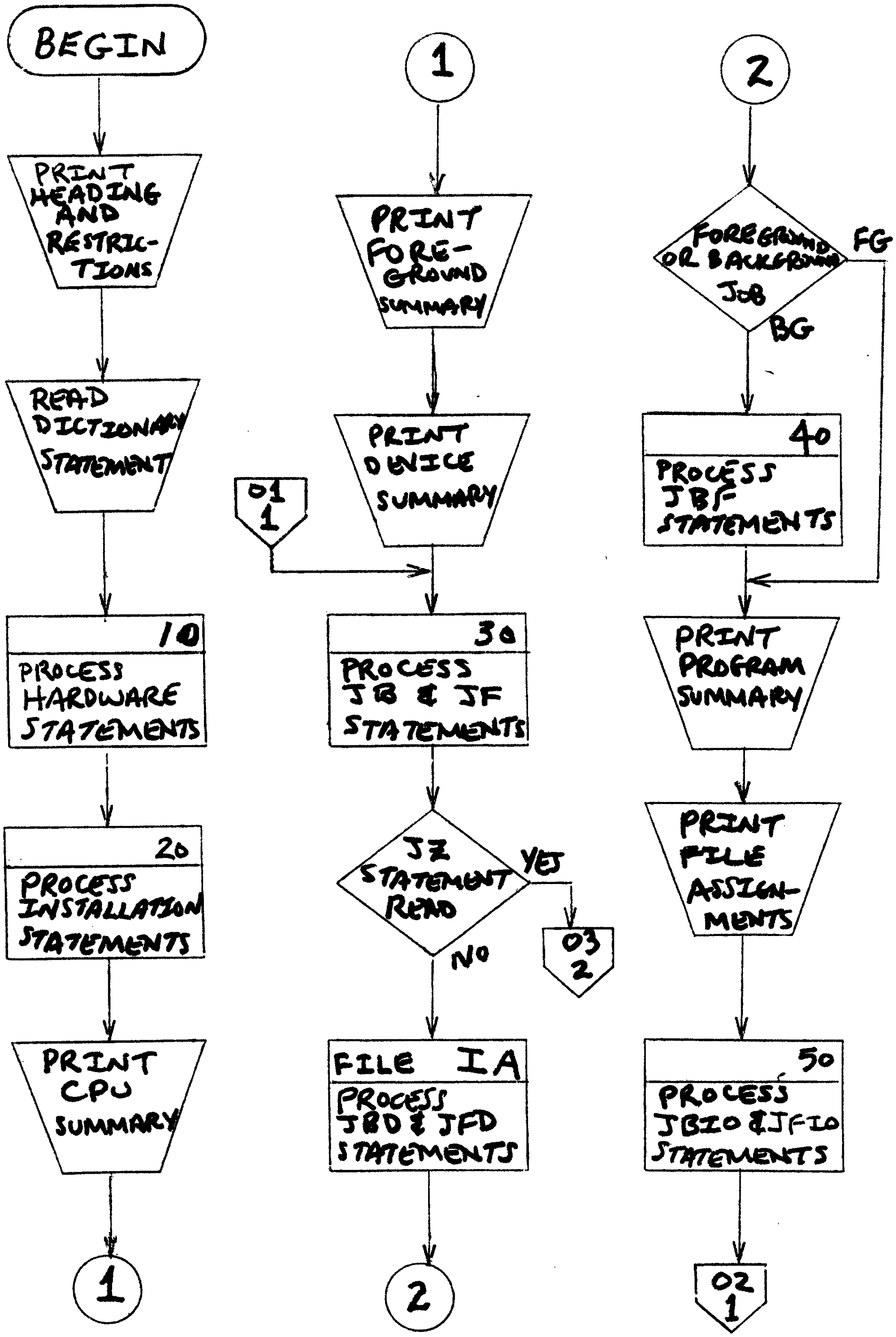
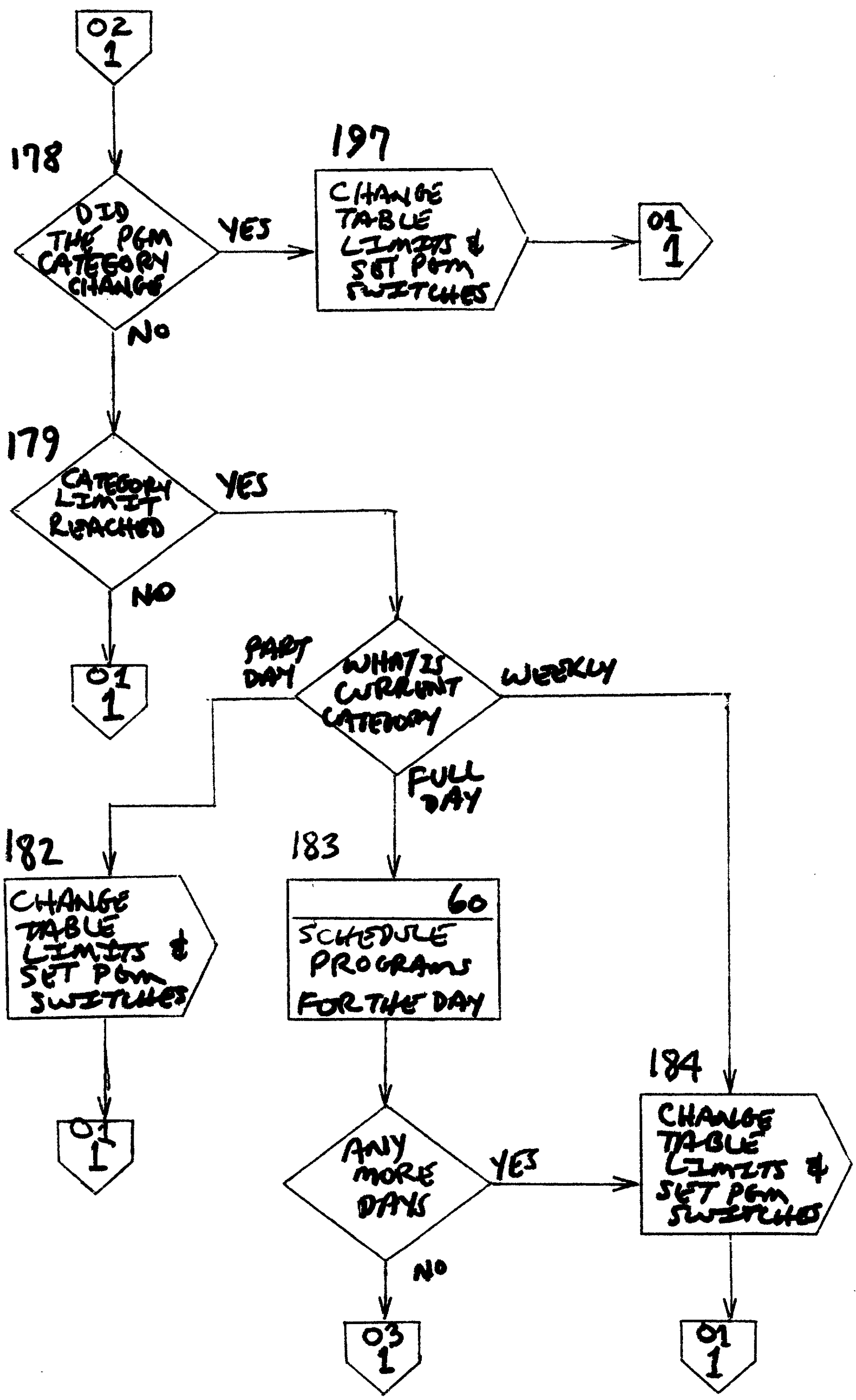


Figure 25. Flowchart symbols.

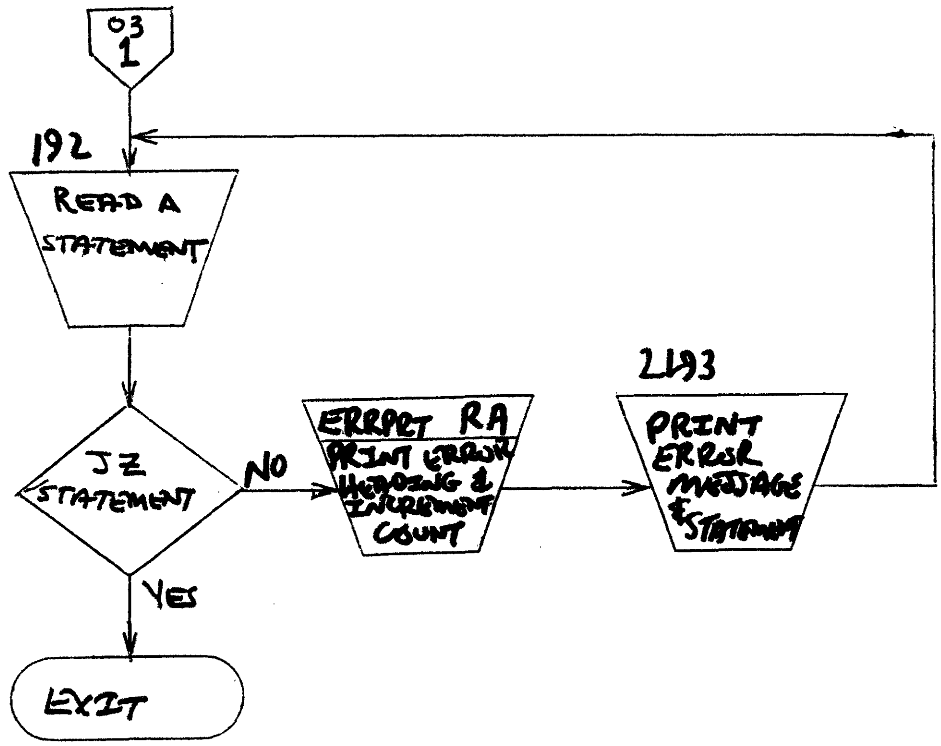
Main Program Chart 0D

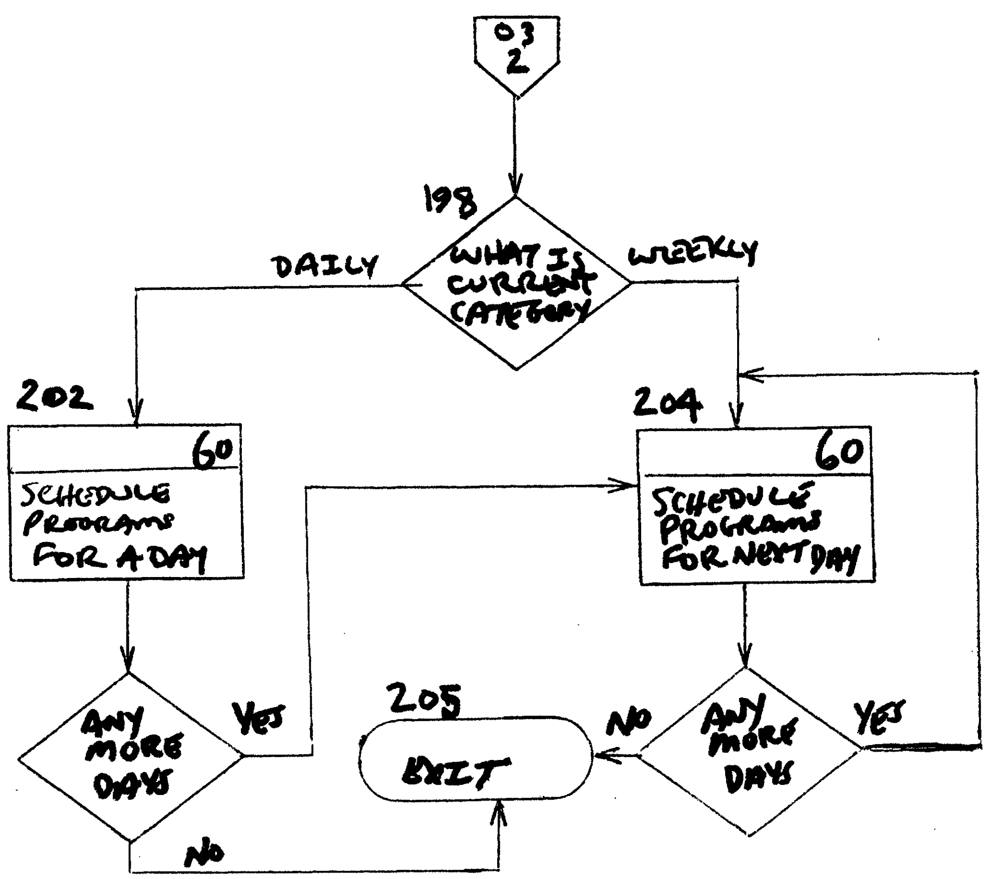



02
1
178
DID THE PGM CATEGORY CHANGE
YES
NO
197
CHANGE TABLE LIMITS & SET PGM SWITCHES
01
1
179
CATEGORY LIMIT REACHED
YES
NO
01
1
WHAT IS CURRENT CATEGORY
PART DAY
WEEKLY
FULL DAY
182
CHANGE TABLE LIMITS & SET PGM SWITCHES
01
1
183
60
SCHEDULE PROGRAMS FOR THE DAY
ANY MORE DAYS
YES
NO
03
1
184
CHANGE TABLE LIMITS & SET PGM SWITCHES
01
1

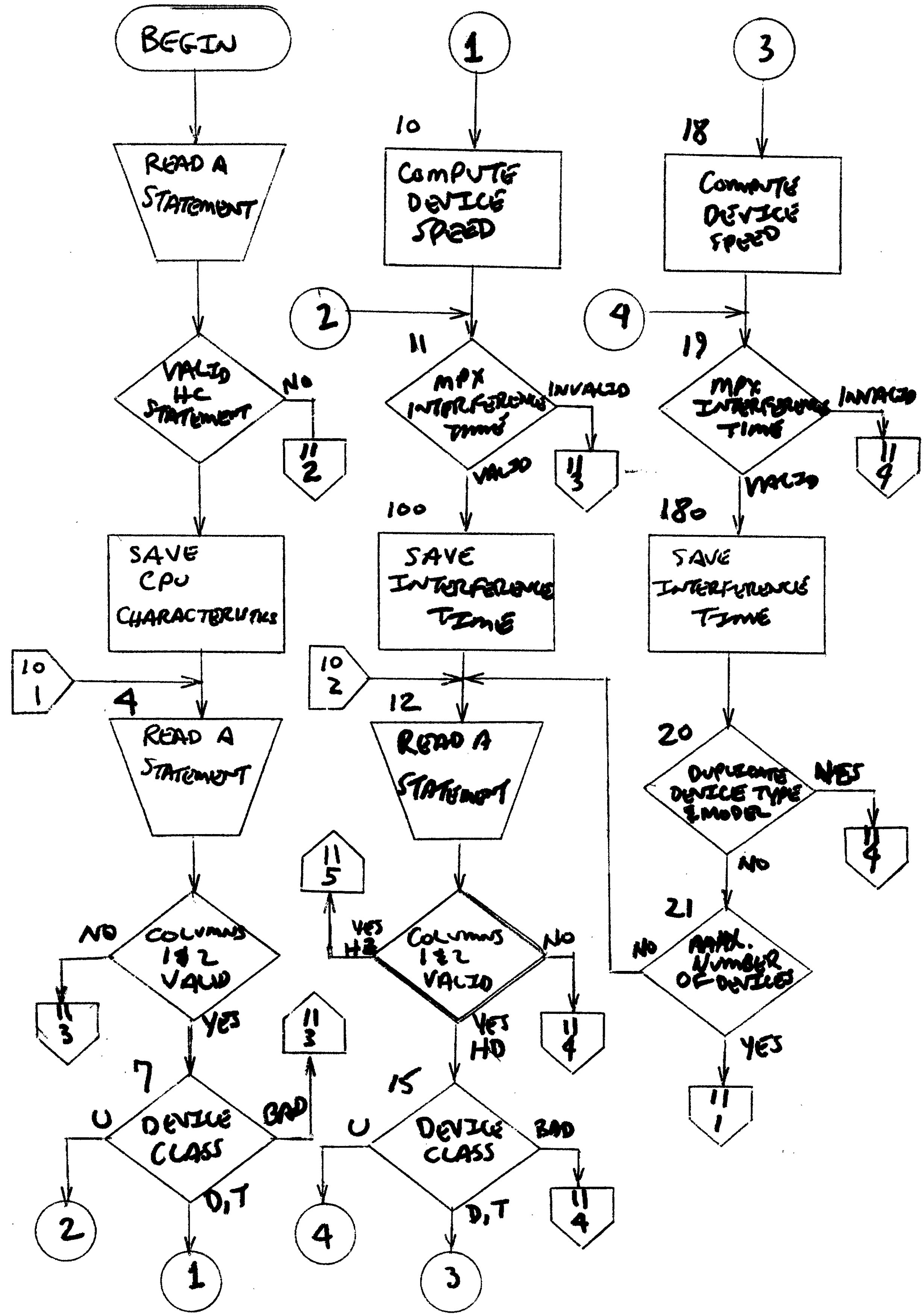
Main Program

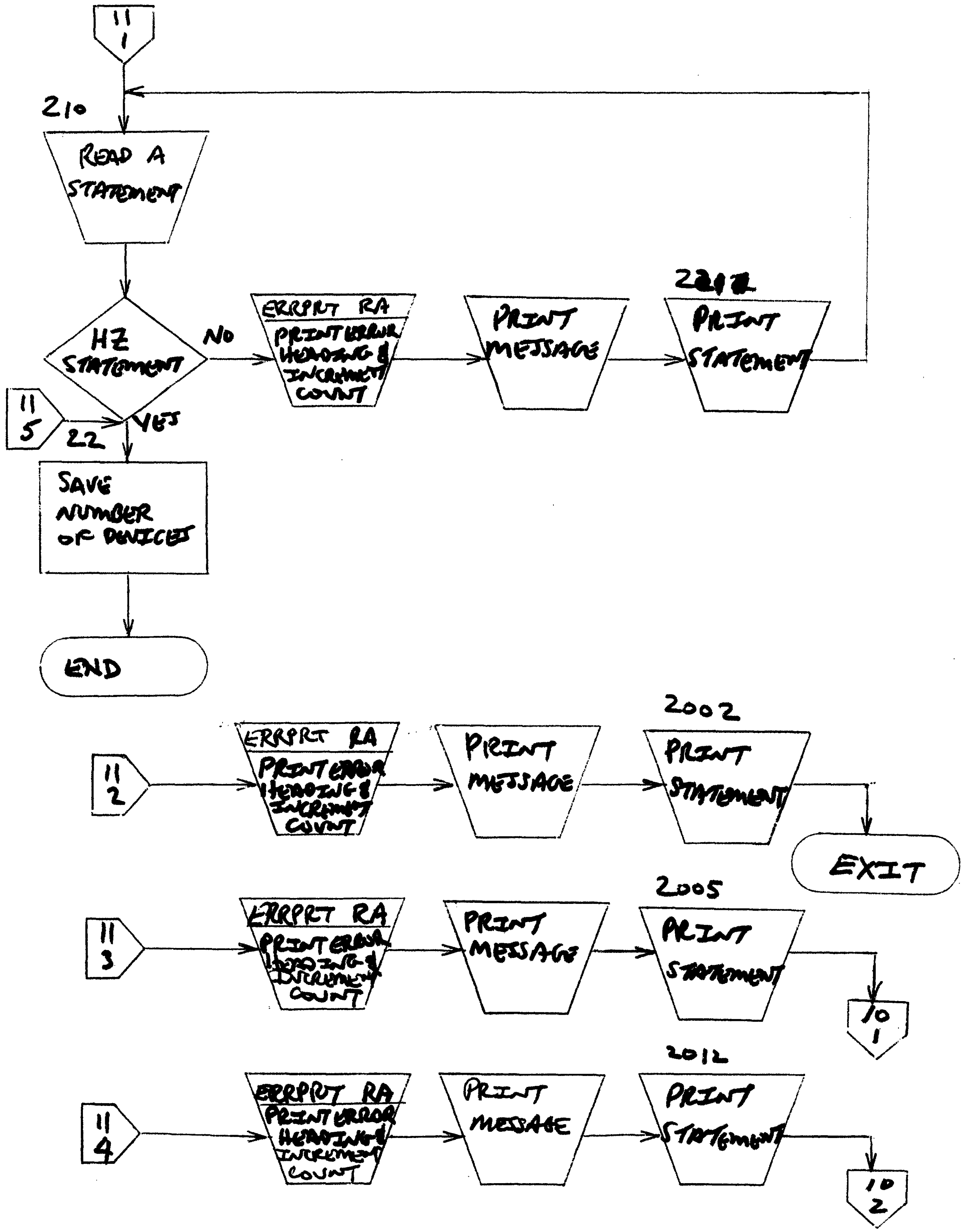
Chart 03

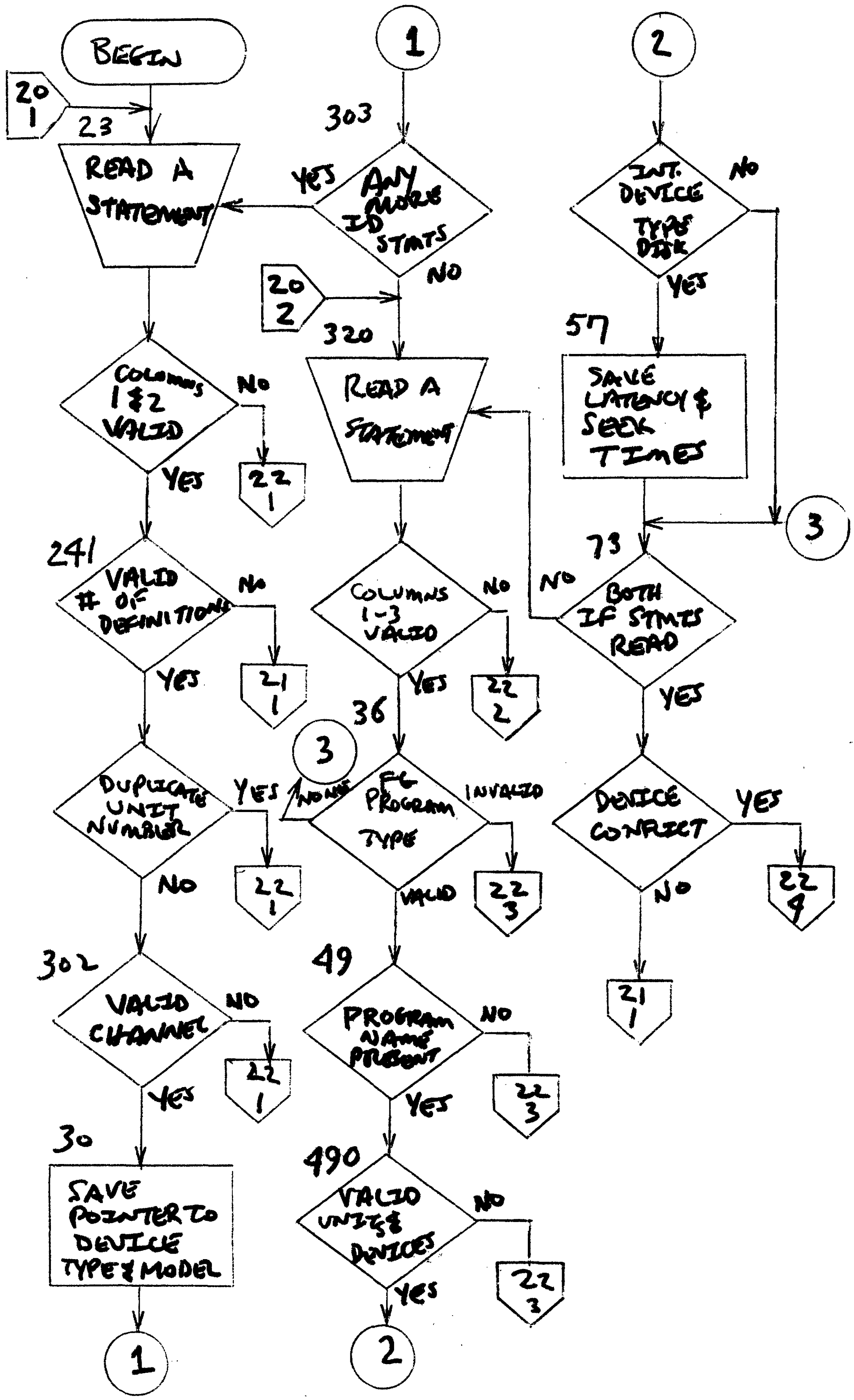




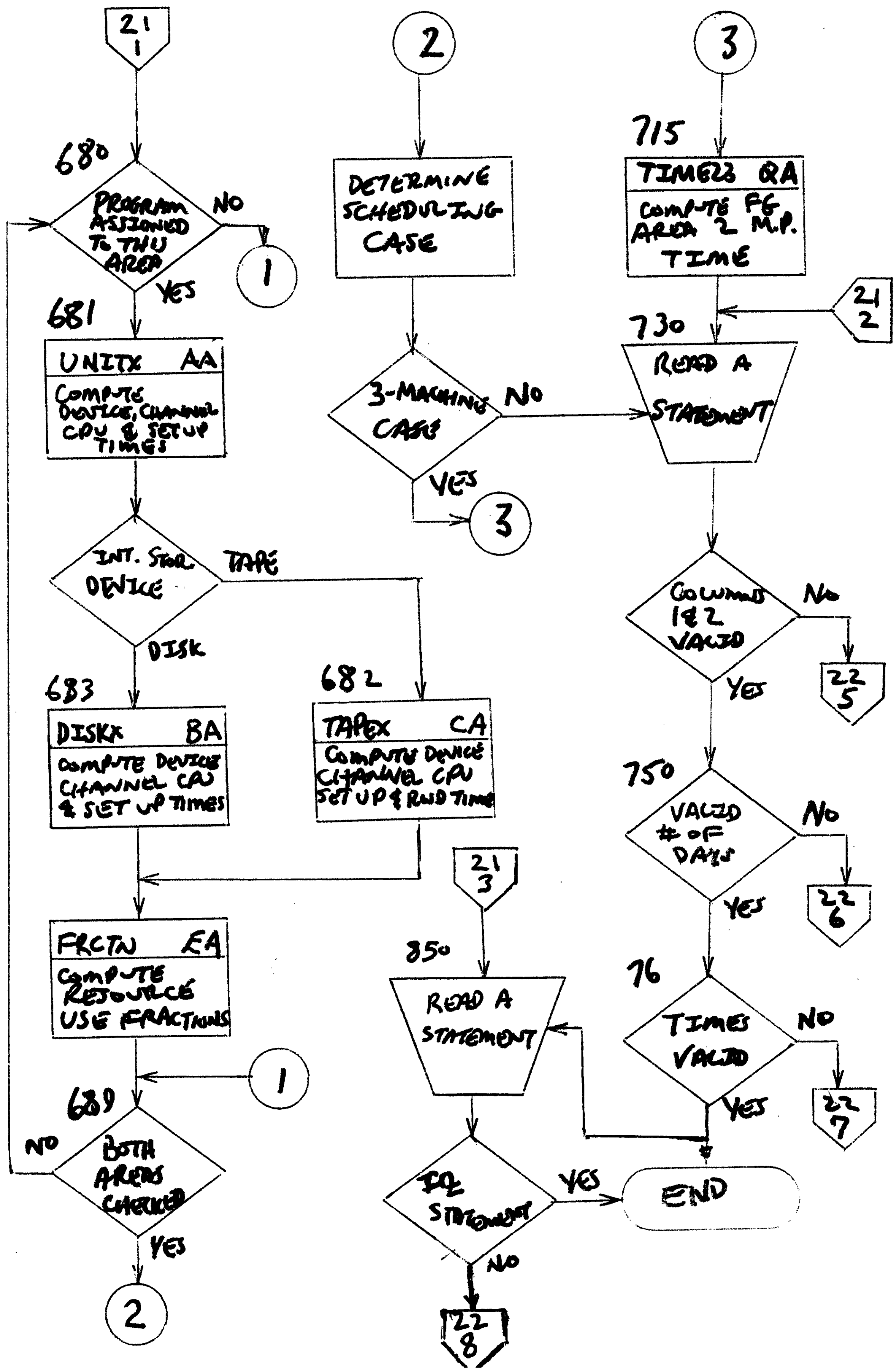
Main Program Chart 10




11
1
210
READ A STATEMENT
HZ STATEMENT
NO
YES
ERRPRT RA
PRINT ERROR HEADING & INCREMENT COUNT
PRINT MESSAGE
2212
PRINT STATEMENT
11
5
22
SAVE NUMBER OF DEVICES
END
11
2
ERRPRT RA
PRINT ERROR HEADING & INCREMENT COUNT
PRINT MESSAGE
2002
PRINT STATEMENT
EXIT
11
3
ERRPRT RA
PRINT ERROR HEADING & INCREMENT COUNT
PRINT MESSAGE
2005
PRINT STATEMENT
10
1
11
4
ERRPRT RA
PRINT ERROR HEADING & INCREMENT COUNT
PRINT MESSAGE
2012
PRINT STATEMENT
10
2

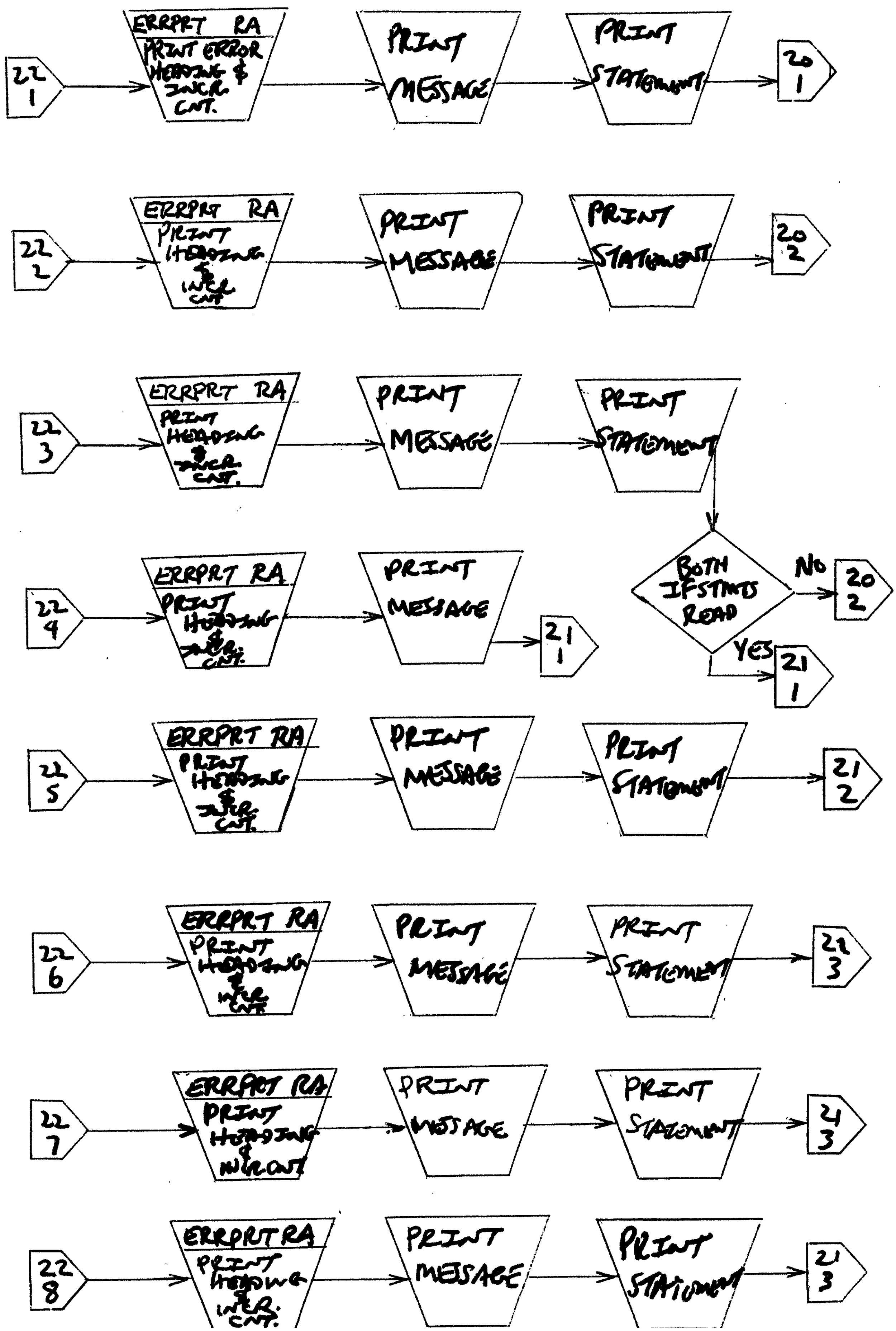

BEGIN
20
1
23
READ A STATEMENT
COLUMNS 1 & 2 VALID
NO
22
1
YES
241
VALID # OF DEFINITIONS
NO
21
1
YES
DUPLICATE UNIT NUMBER
YES
22
1
NO
302
VALID CHANNEL
NO
22
1
YES
30
SAVE POINTER TO DEVICE TYPE & MODEL
1
1
303
ANY MORE ID STMTS
YES
NO
20
2
320
READ A STATEMENT
COLUMNS 1-3 VALID
NO
22
2
YES
36
3
NONE
FG PROGRAM TYPE
INVALID
22
3
VALID
49
PROGRAM NAME PRESENT
NO
22
3
YES
490
VALID UNITS & DEVICES
NO
22
3
YES
2
2
INT. DEVICE TYPE DISK
NO
YES
57
SAVE LATENCY & SEEK TIMES
3
73
BOTH IF STMTS READ
NO
YES
DEVICE CONFLICT
YES
22
4
NO
21
1

Main Program Chart 21

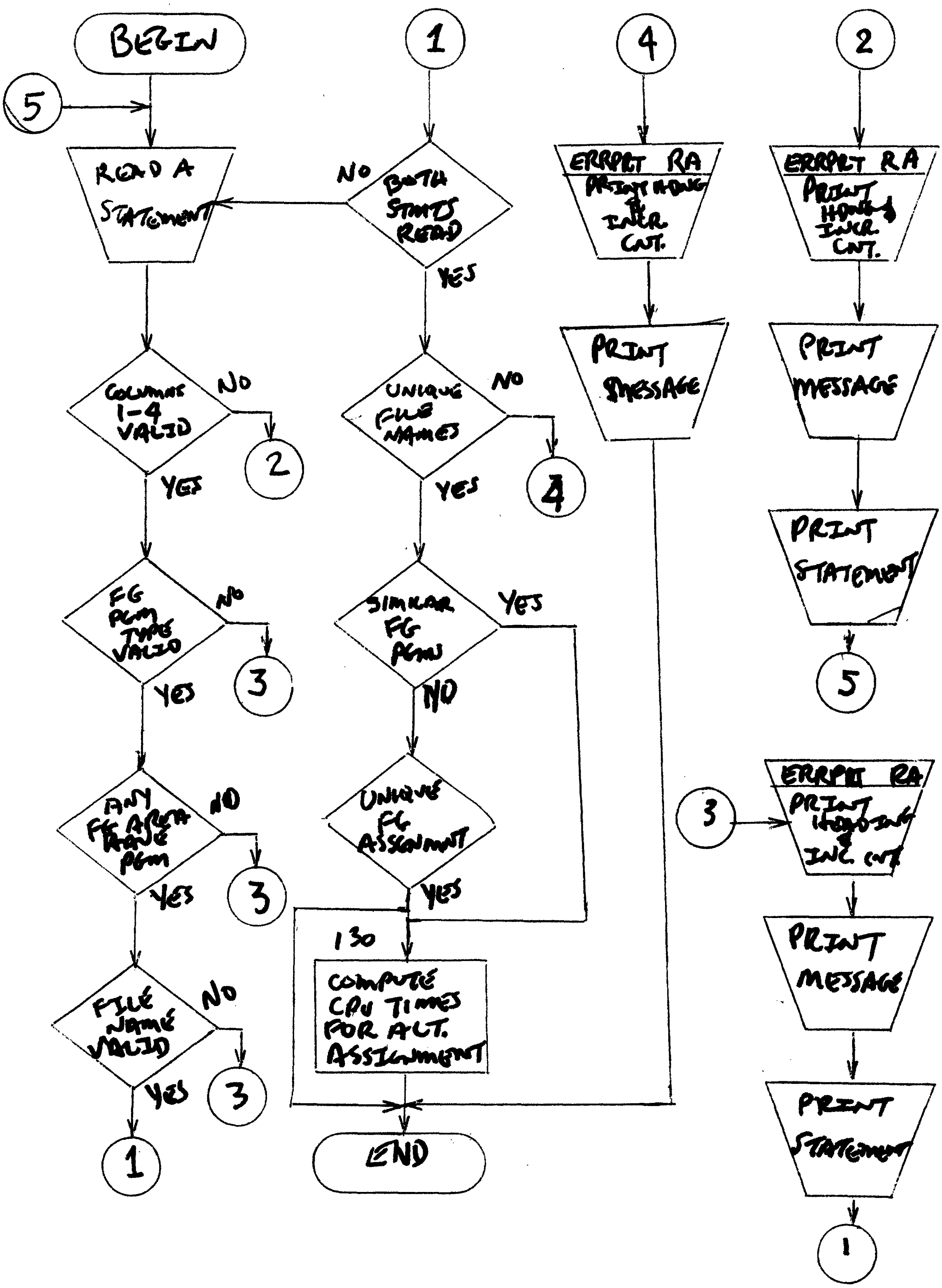


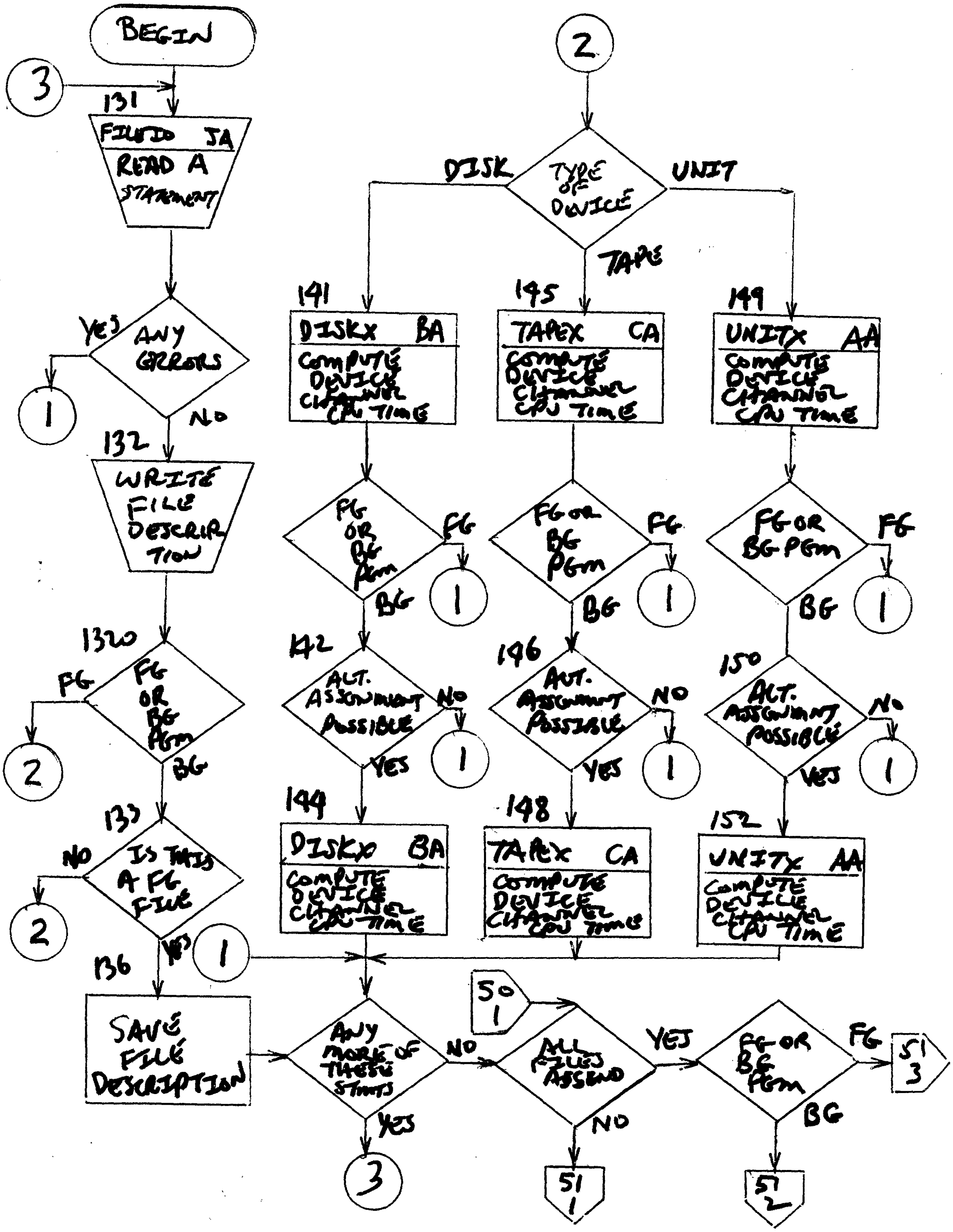
Main Program

Chart 22



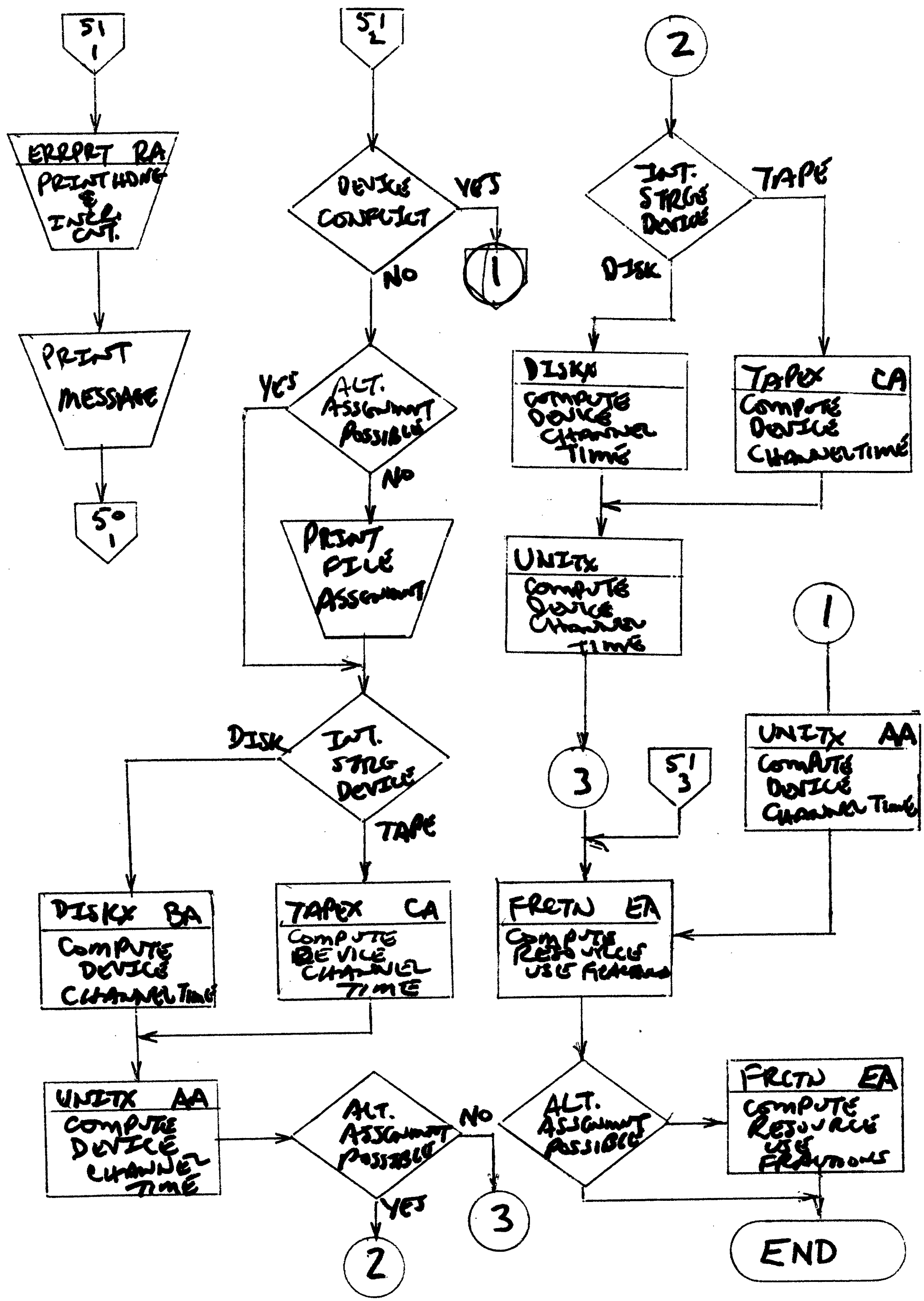
Main Program Chart 40



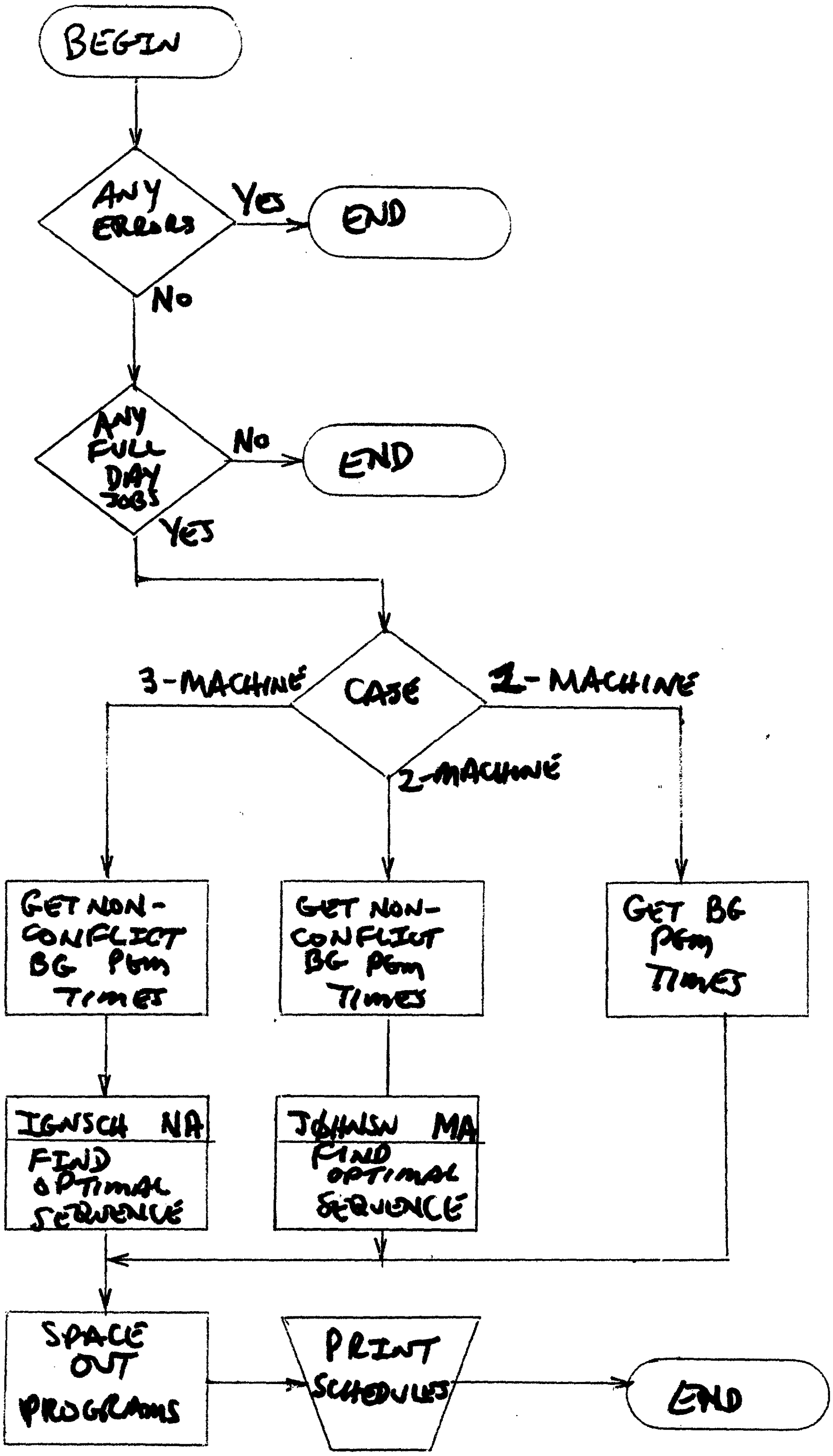

BEGIN
131
FILEID JA
READ A STATEMENT
ANY ERRORS
YES
NO
132
WRITE FILE DESCRIPTION
1320
FG OR BG PGM
FG
BG
133
IS THIS A FG FILE
NO
YES
136
SAVE FILE DESCRIPTION
TYPE OF DEVICE
DISK
TAPE
UNIT
141
DISKX BA
COMPUTE DEVICE CHANNEL CPU TIME
145
TAPEX CA
COMPUTE DEVICE CHANNEL CPU TIME
149
UNITX AA
COMPUTE DEVICE CHANNEL CPU TIME
FG OR BG PGM
142
ALT. ASSIGNMENT POSSIBLE
146
150
144
DISKX BA
148
TAPEX CA
152
UNITX AA
ANY MORE OF THESE STMTS
ALL FILES ASSGND
50 1
51 1
51 2
51 3

Main Program

Chart 51

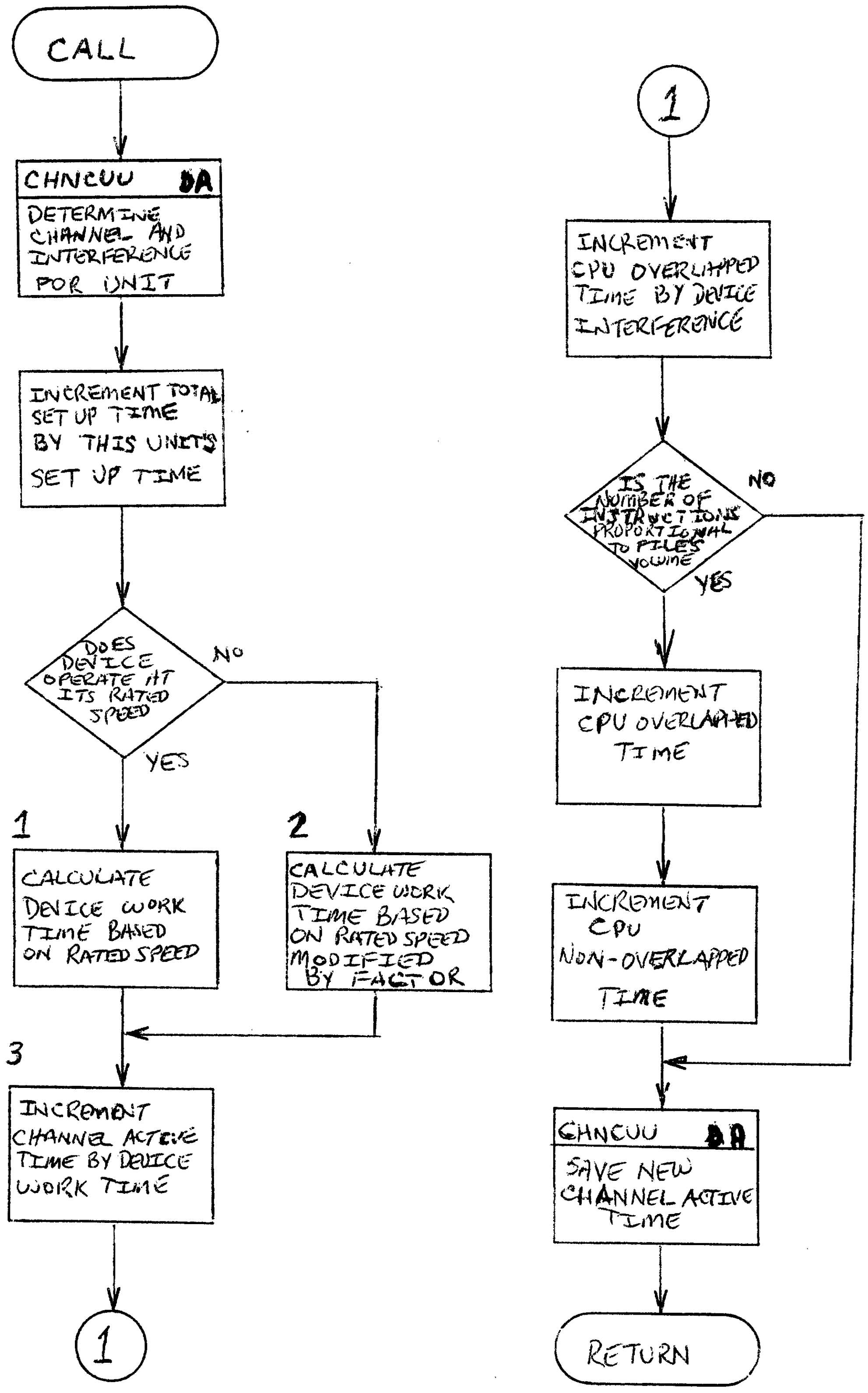


Main Program Chart 60

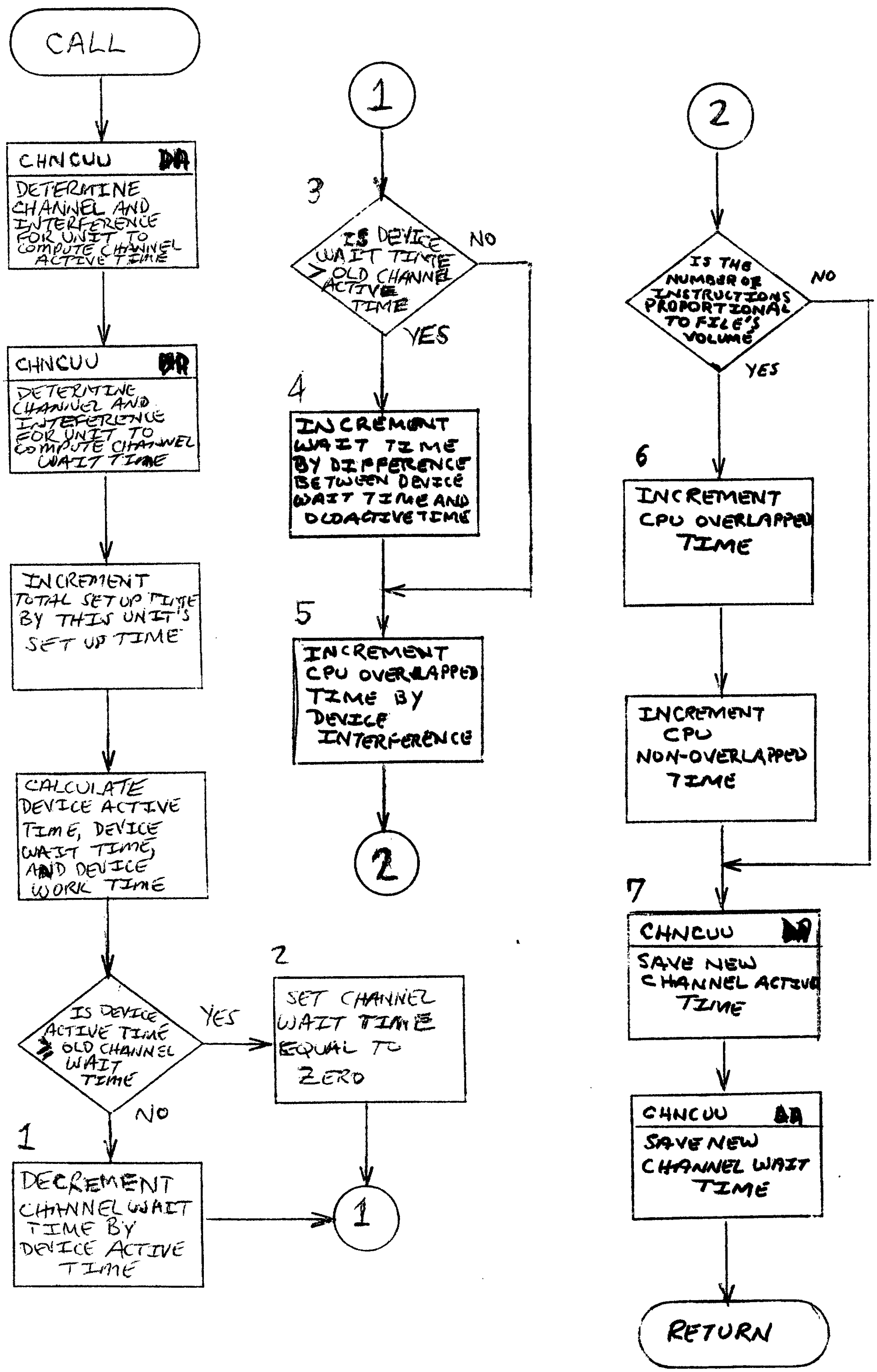


Subprogram UNITX

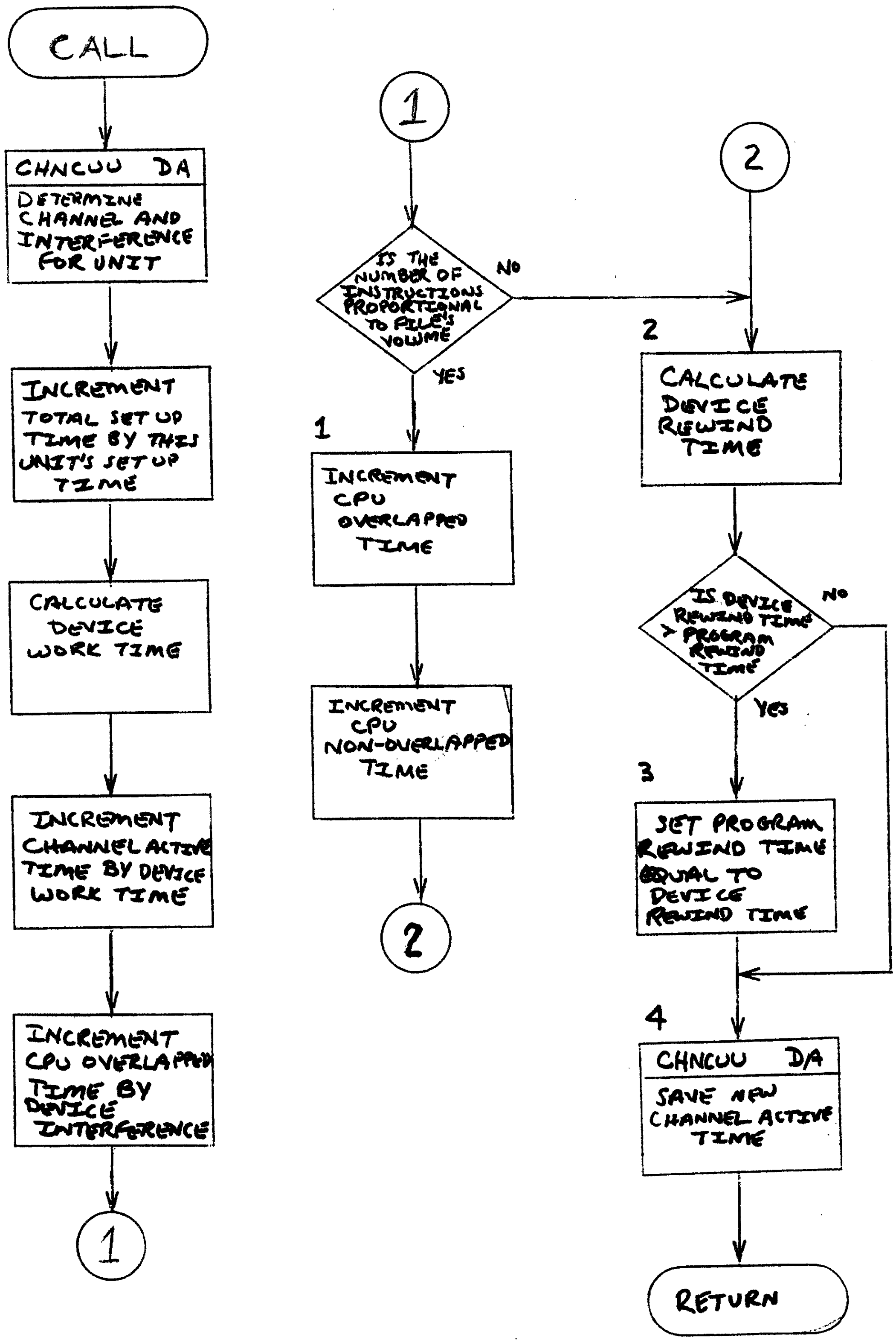
Chart AA



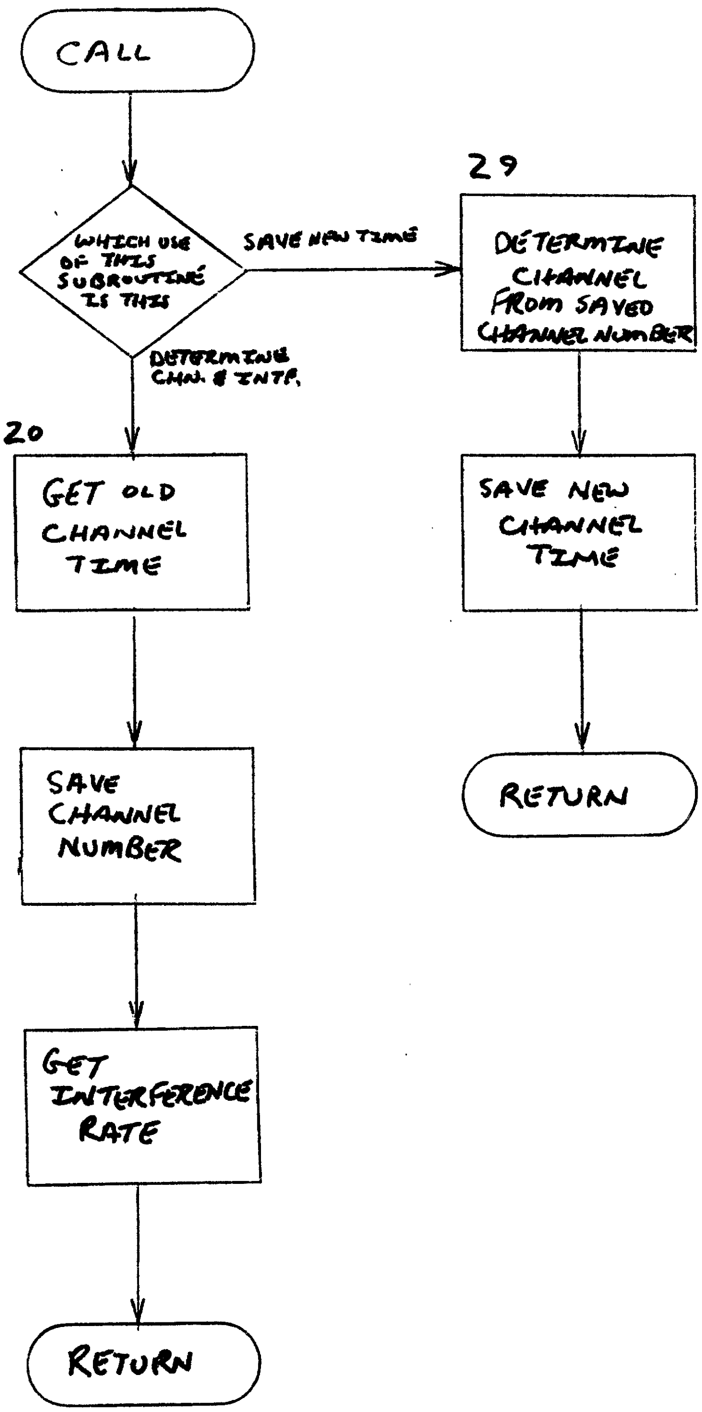
Subprogram DISKX Chart BA



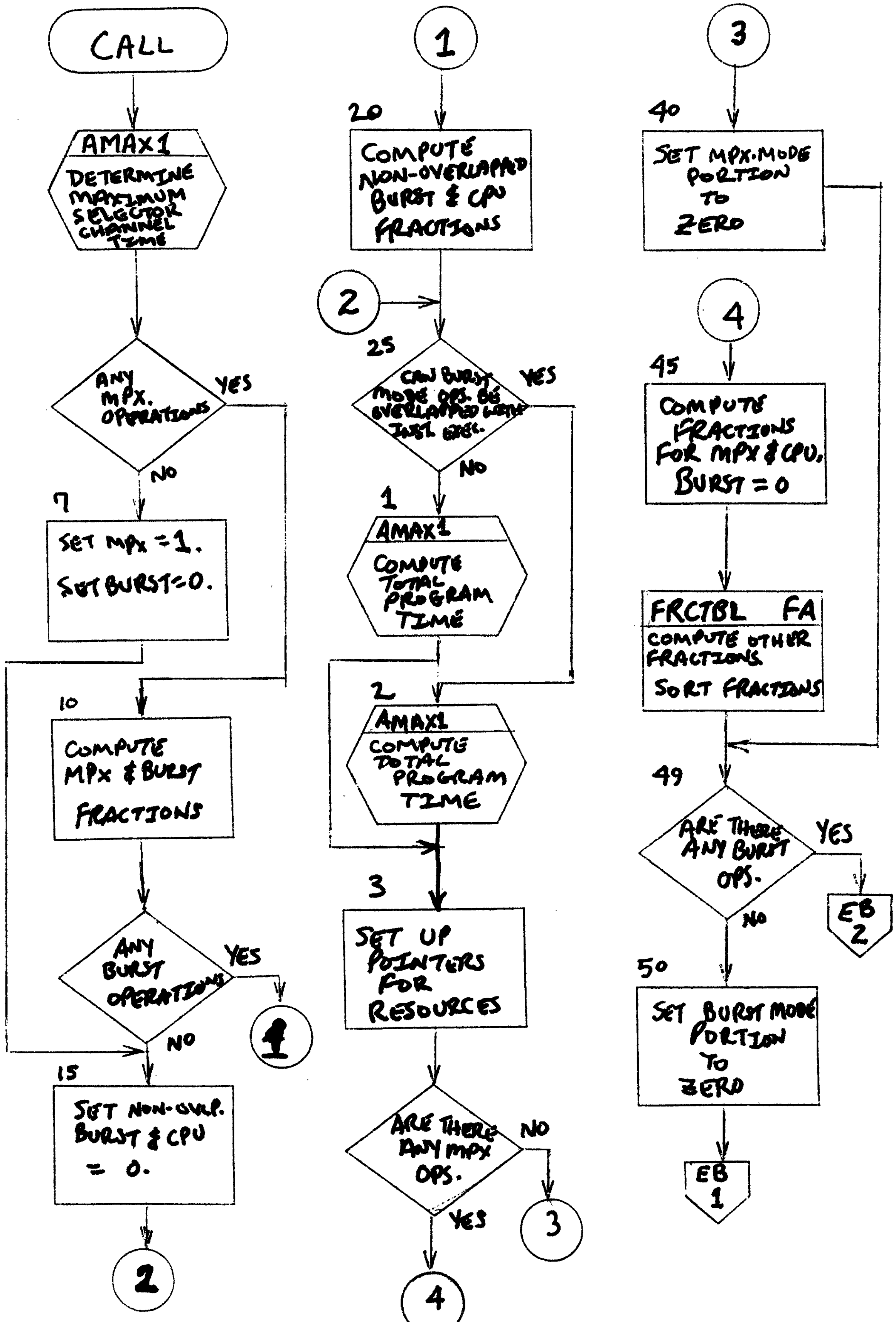
Subprogram TAPEX Chart CA



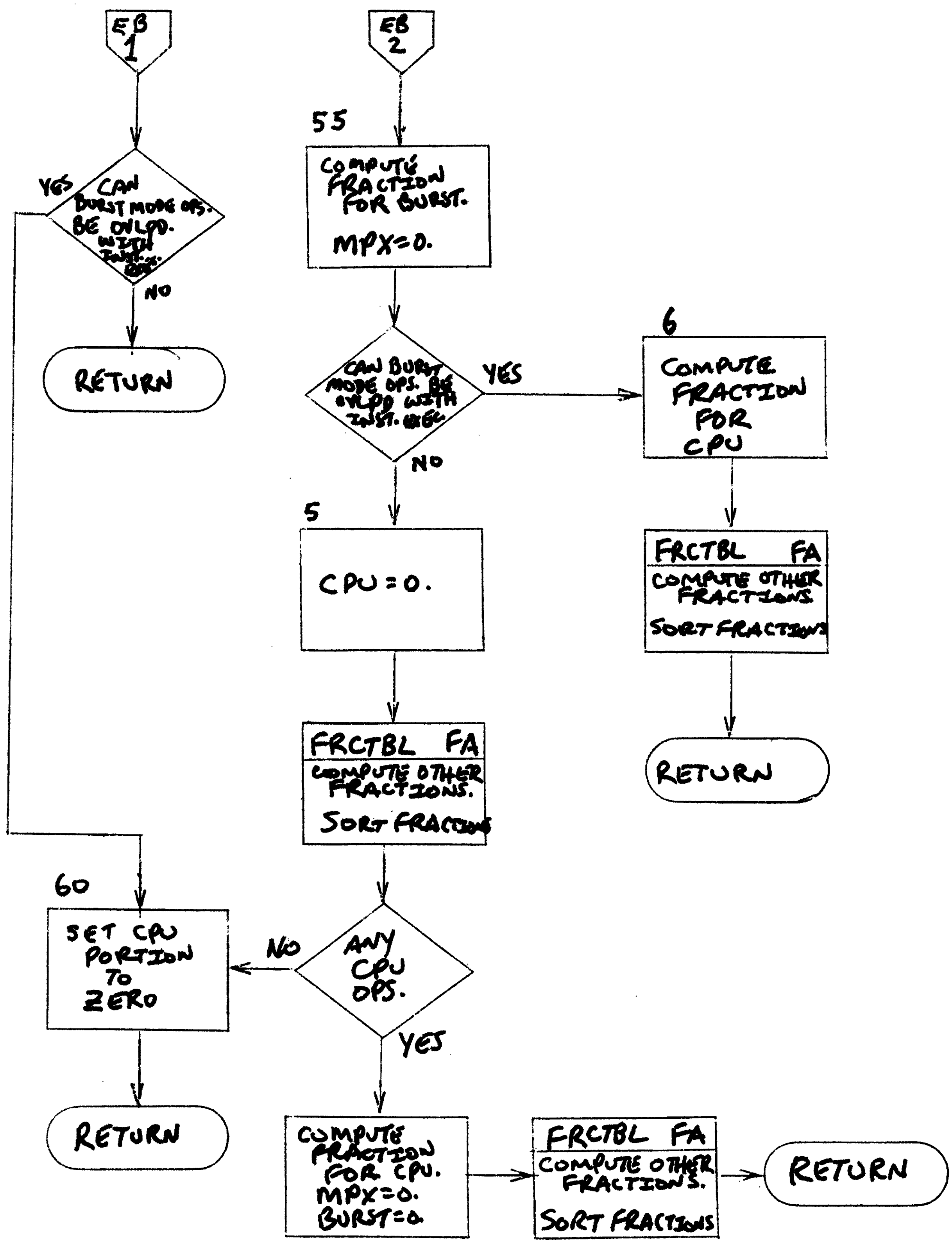
Subprogram CHNCUU Chart DA



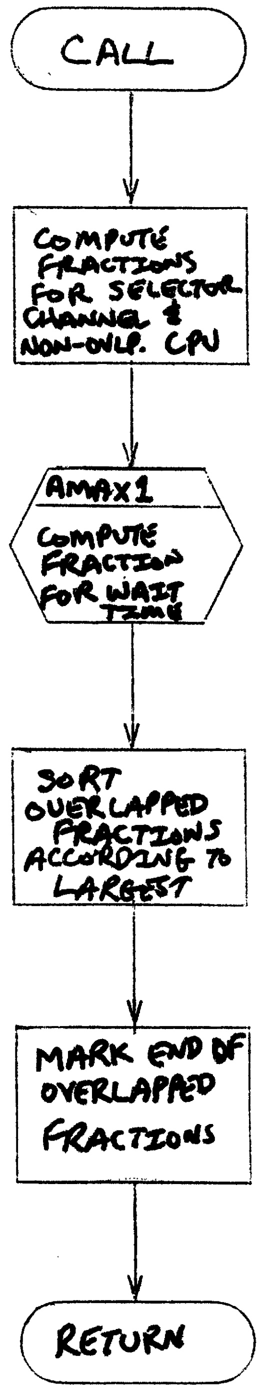
Subprogram FRCTN Chart EA



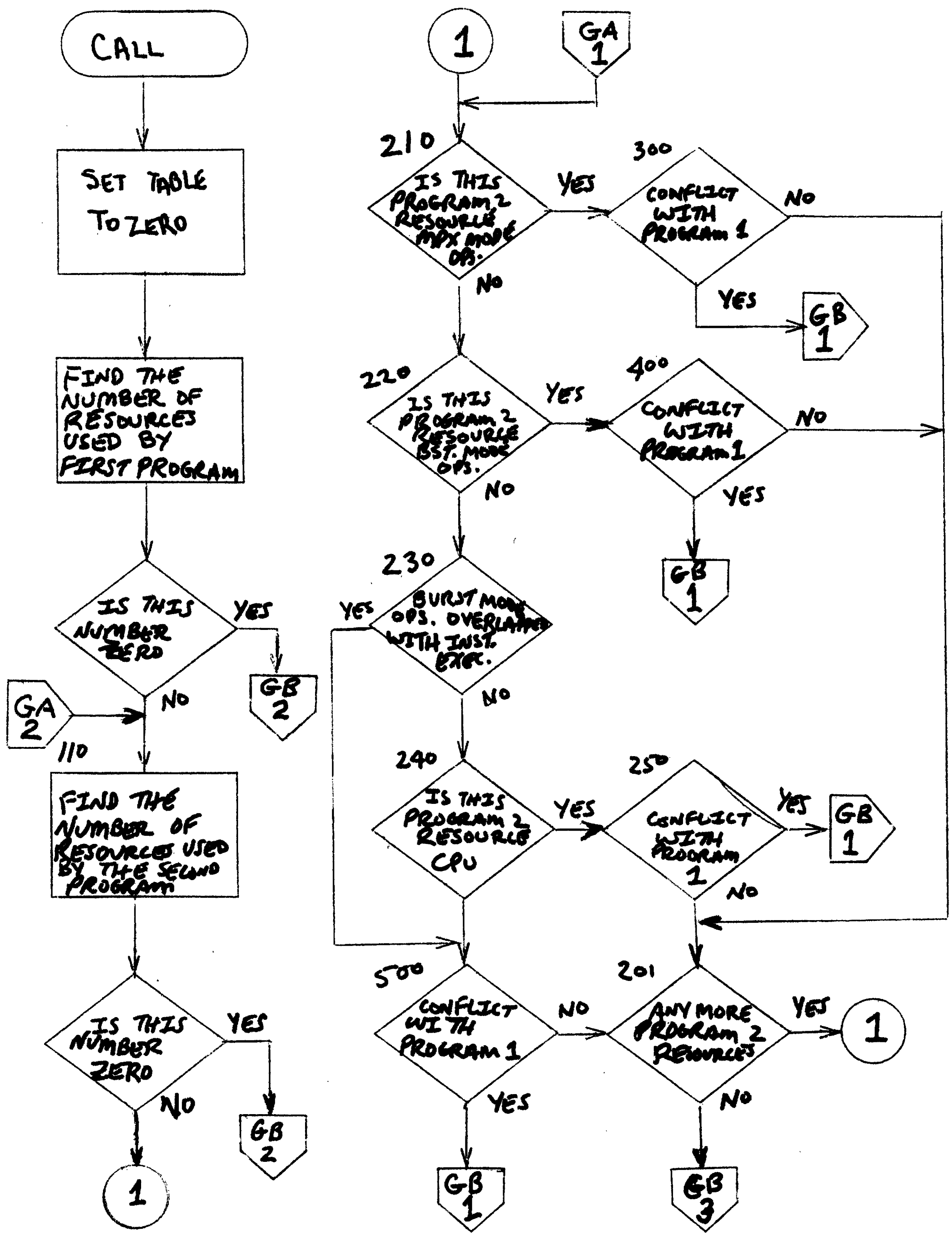
Subprogram FRCTN Chart EB



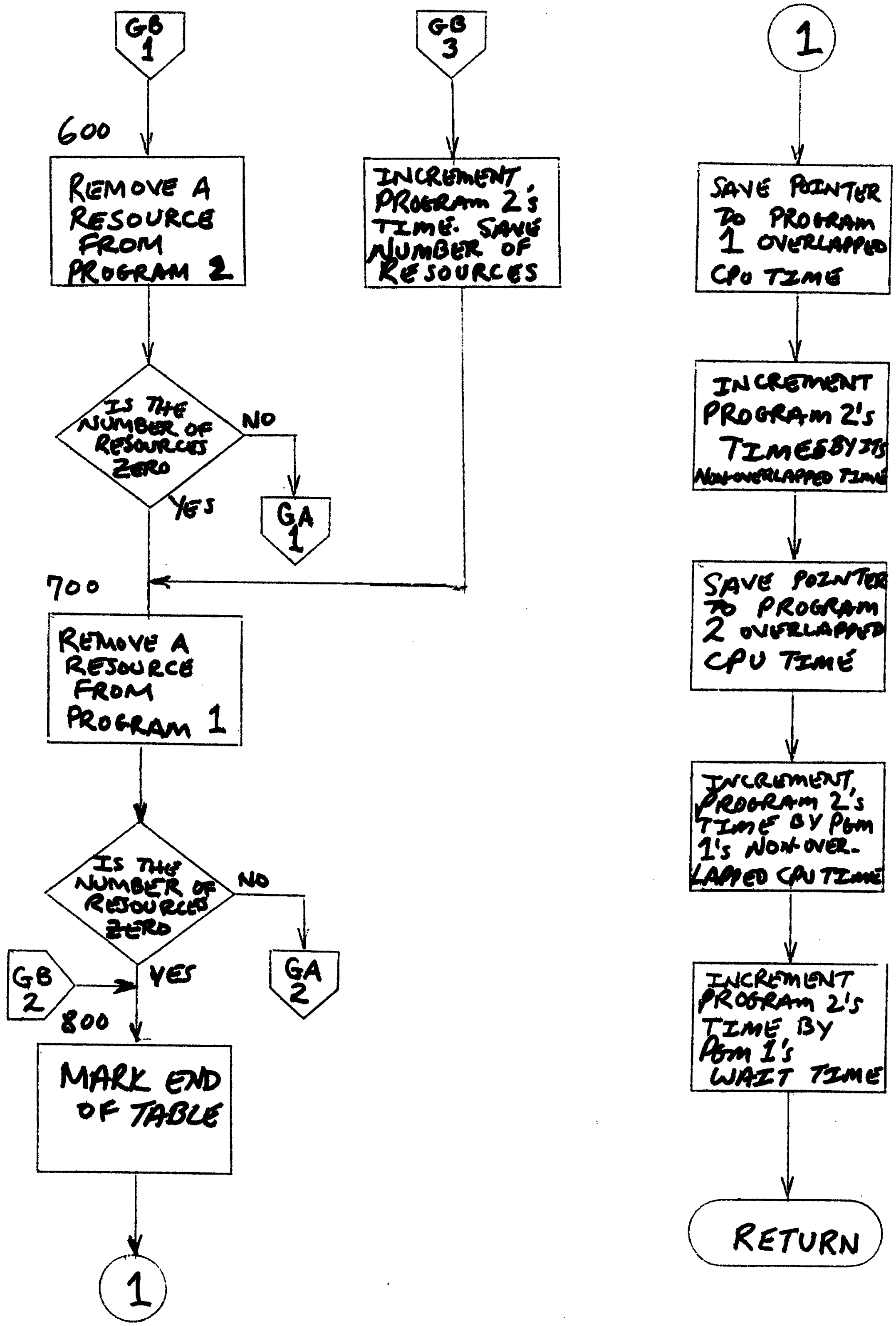
Subprogram FRCTBL Chart FA



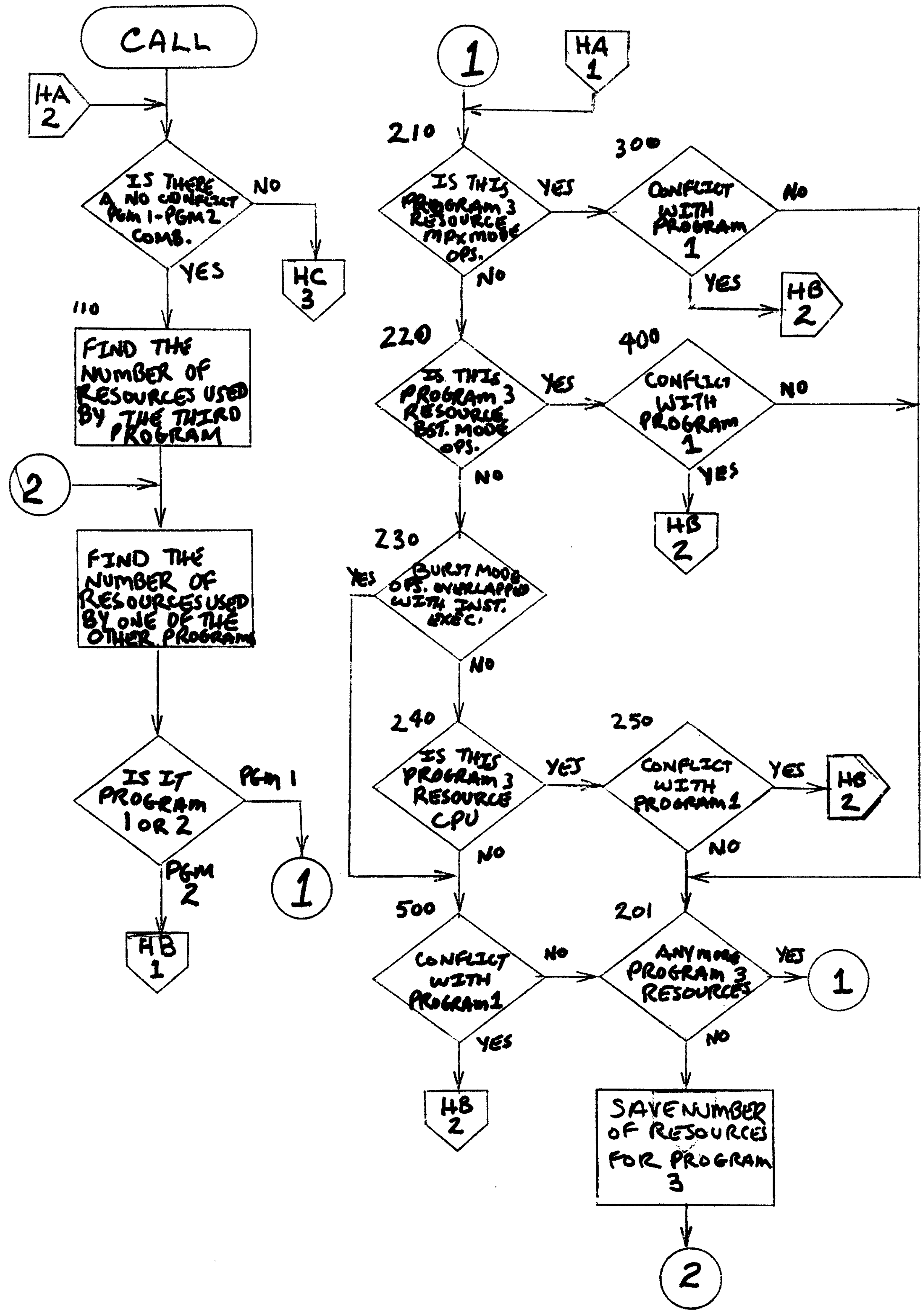
Subprogram TIME2 Chart GA



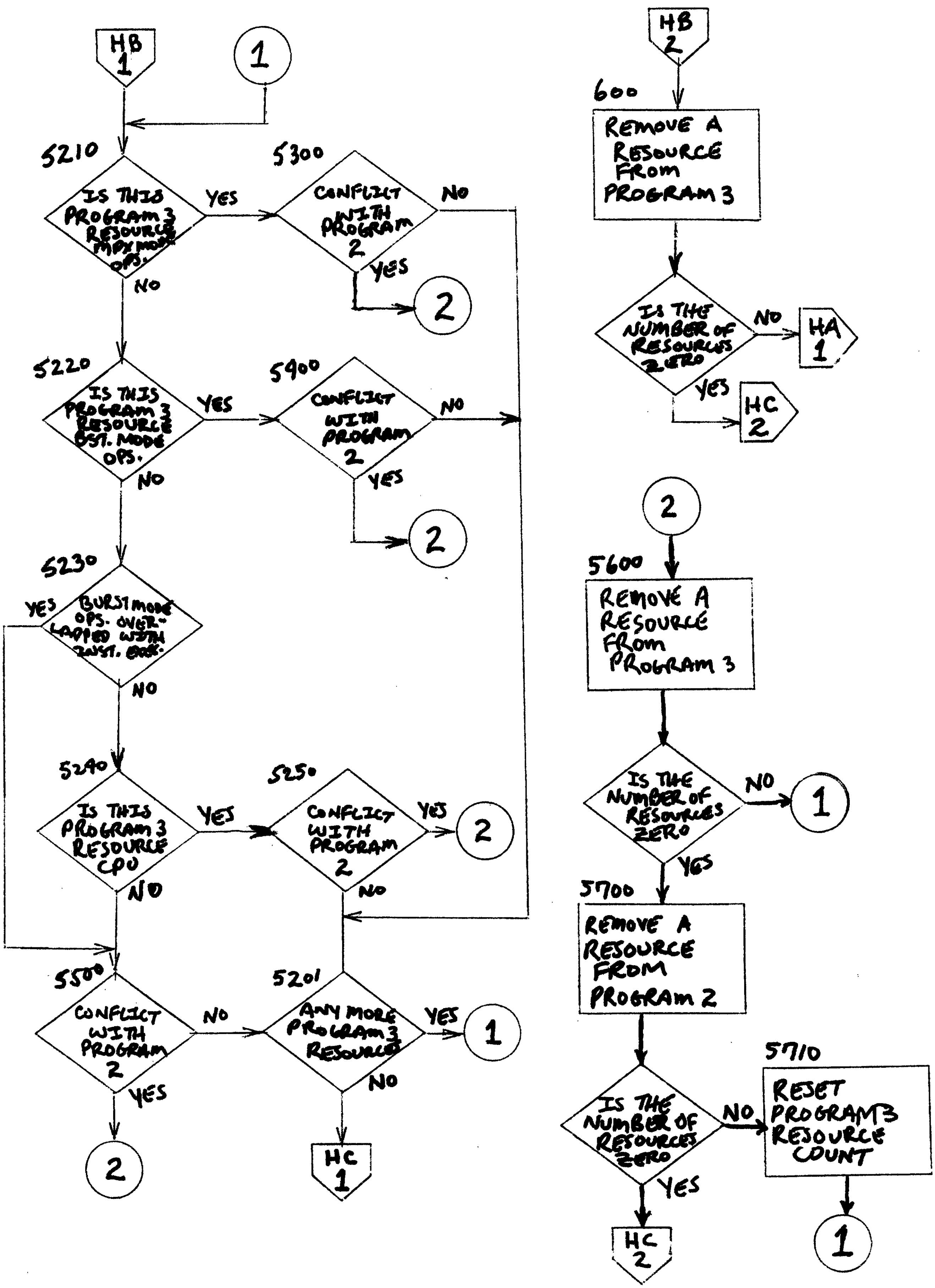
Subprogram TIME2 Chart GB



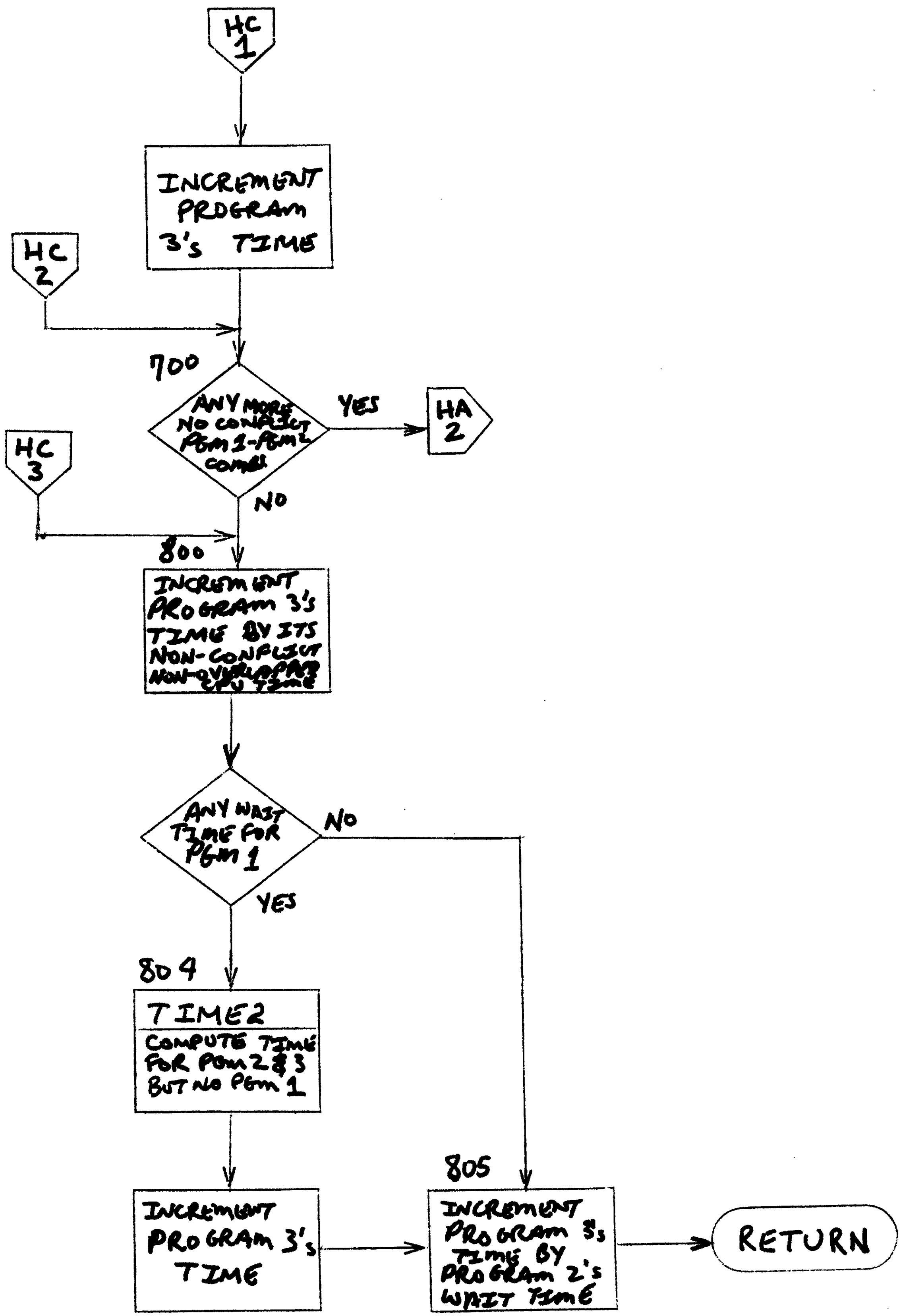
Subprogram TIME3 Chart HA



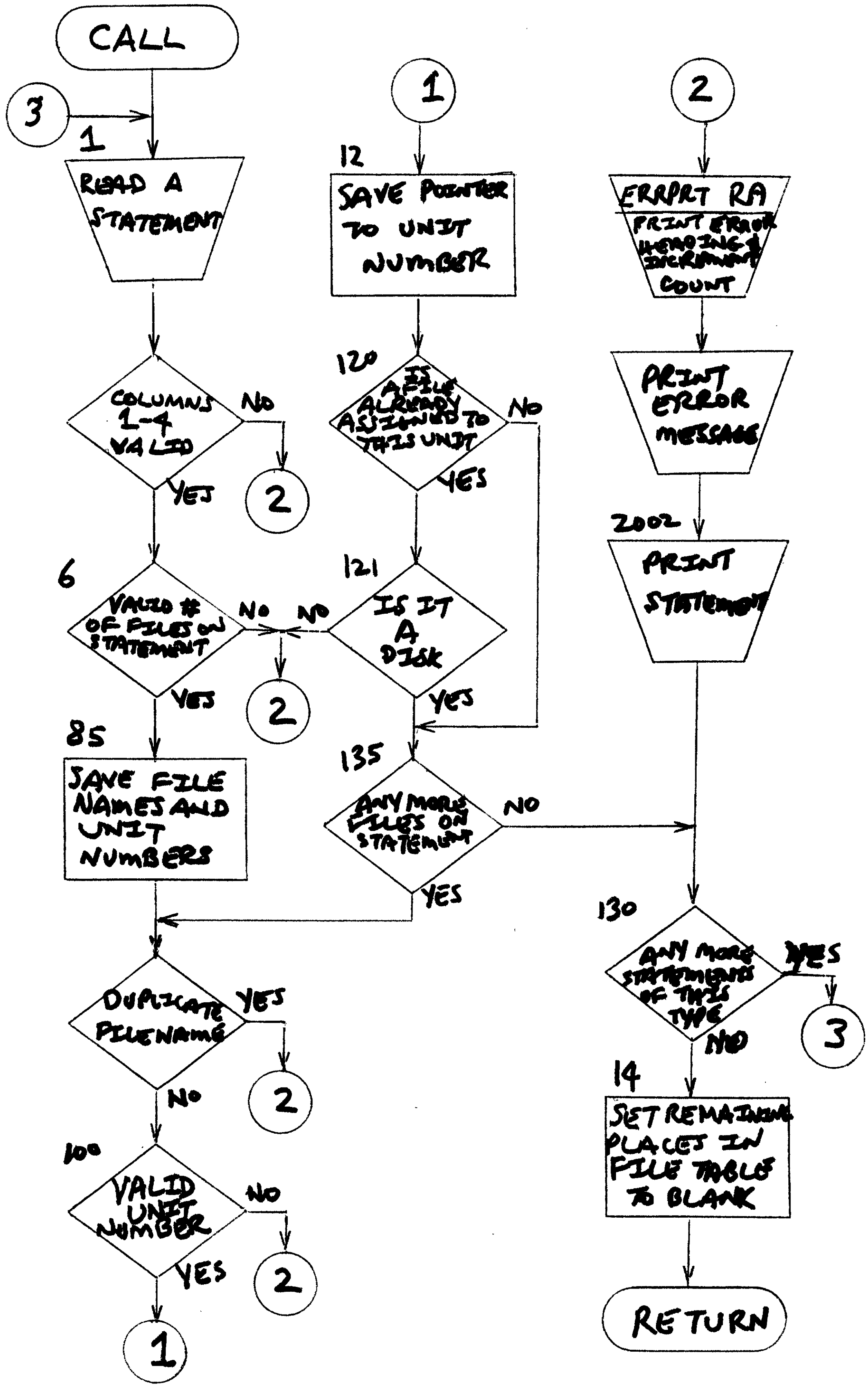
Subprogram TIME3 Chart HB



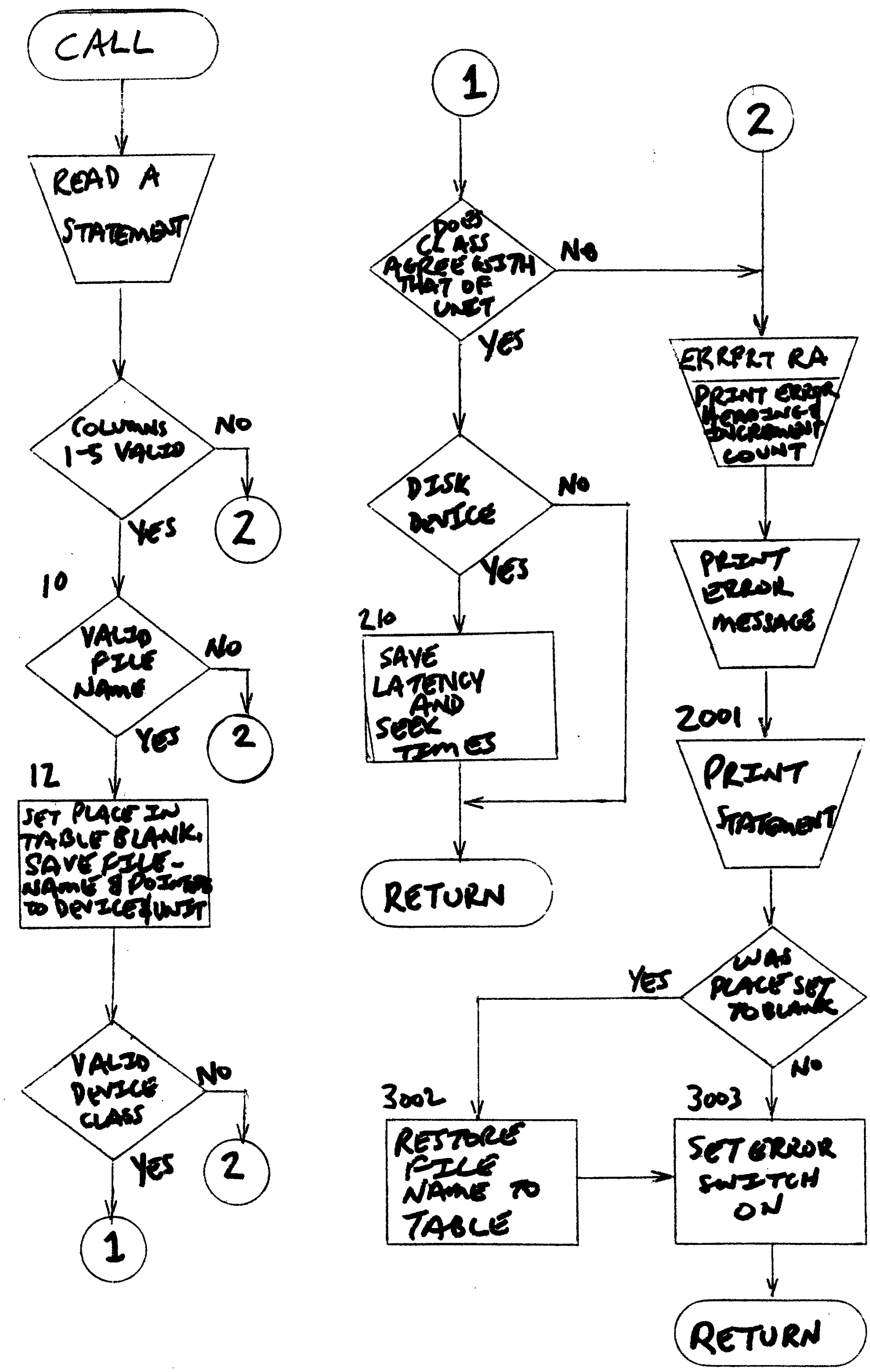
Subprogram TIME3 Chart HC



Subprogram FILE   Chart IA

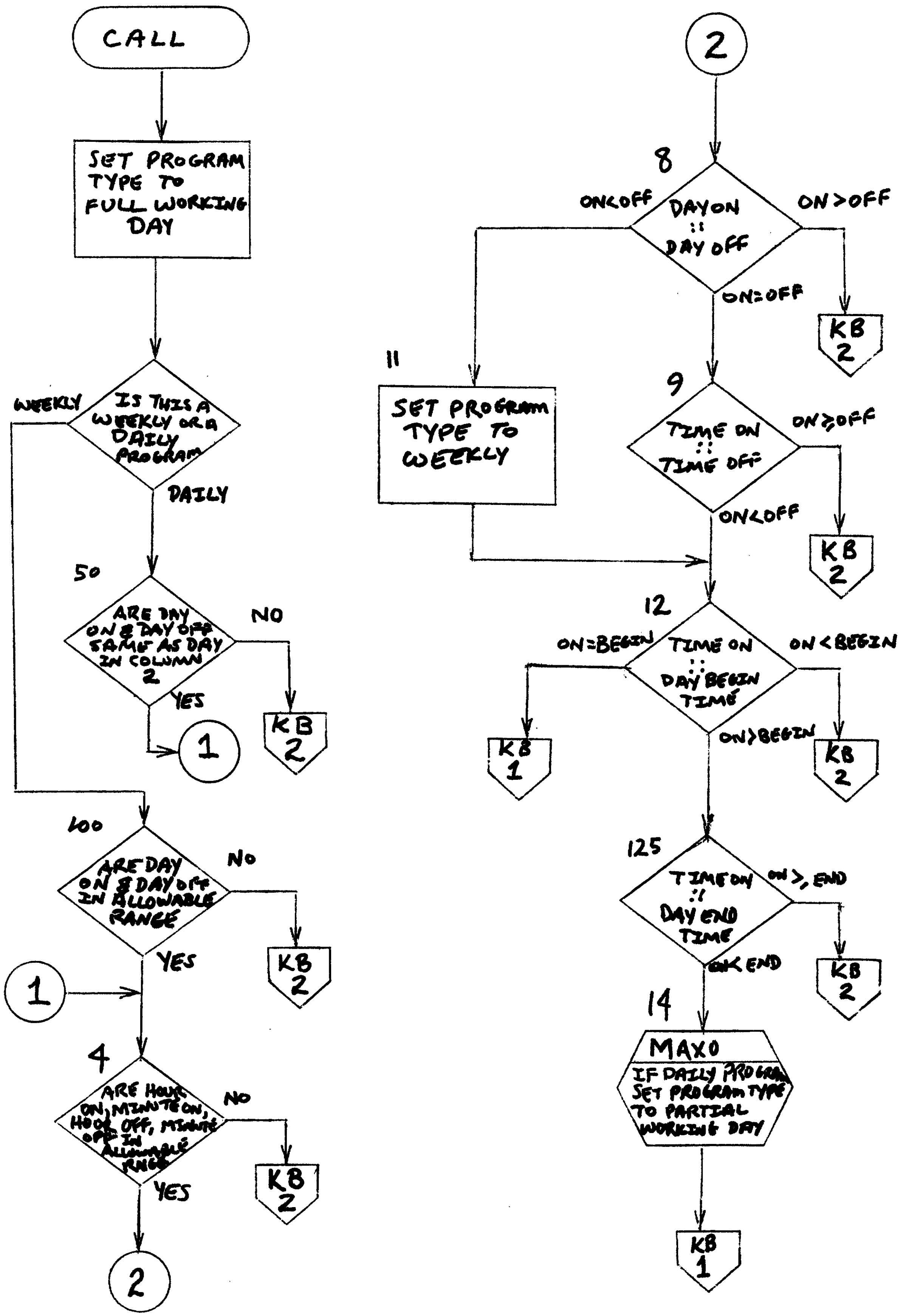


Subprogram FILEIO Chart JA

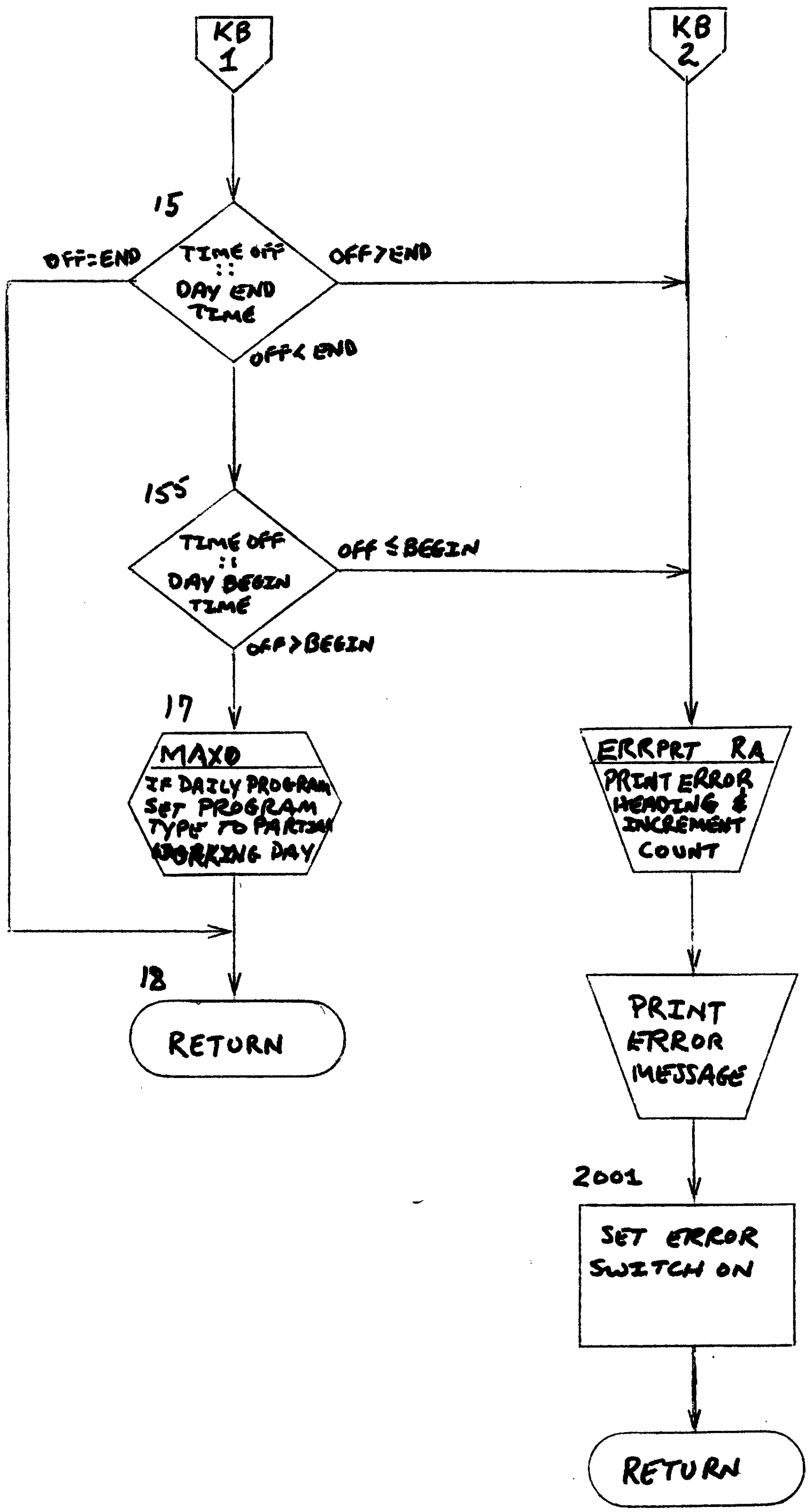


Subprogram TOD

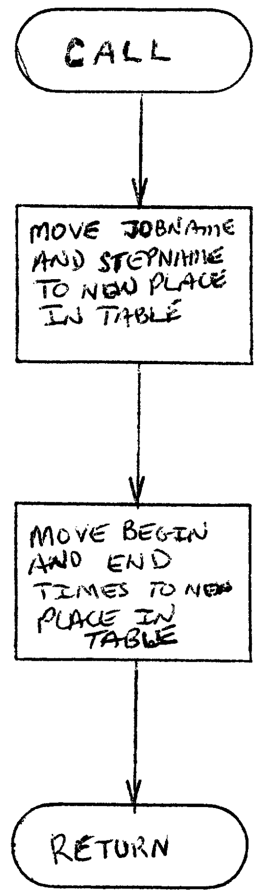
Chart KA



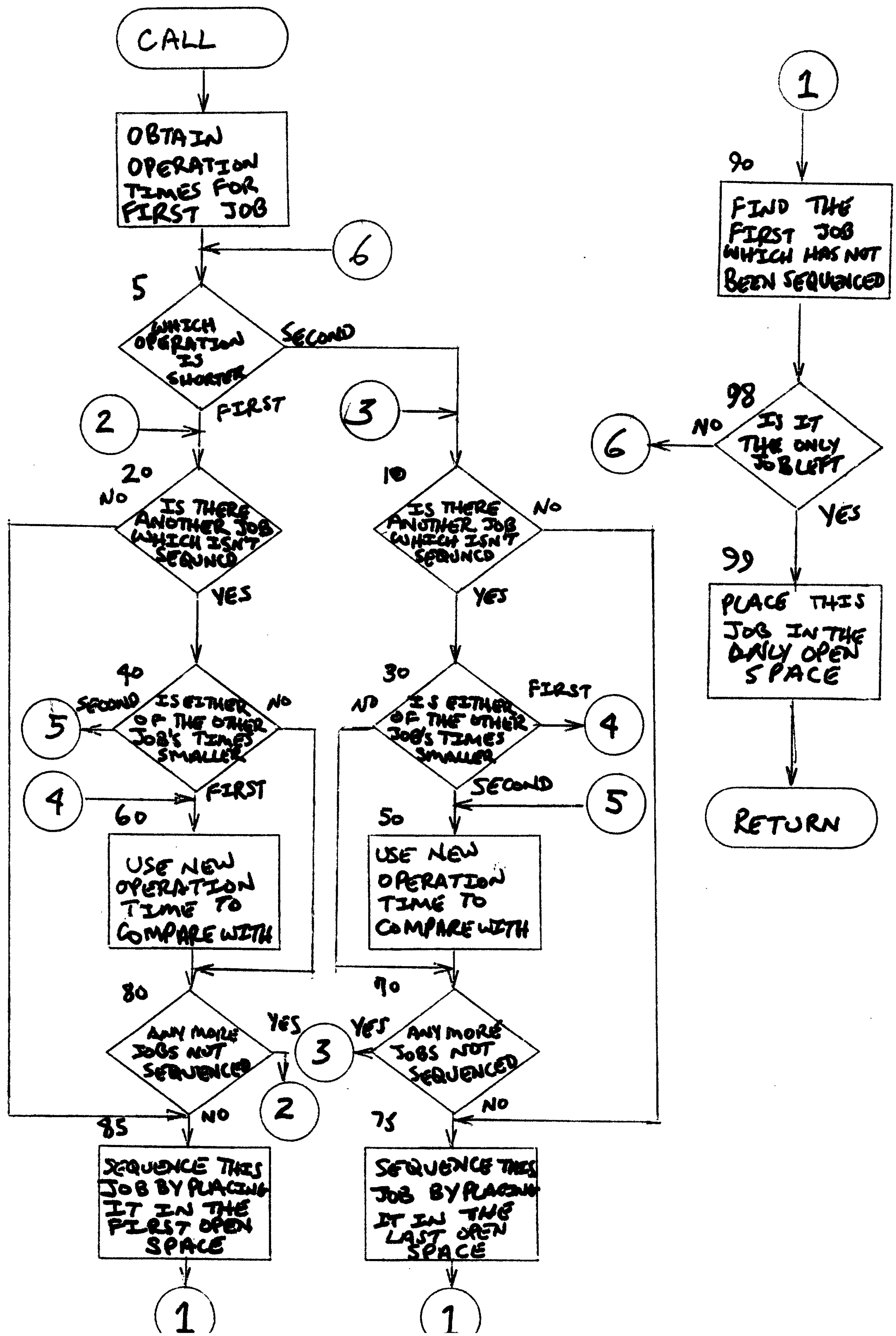
Subprogram TOD Chart KB



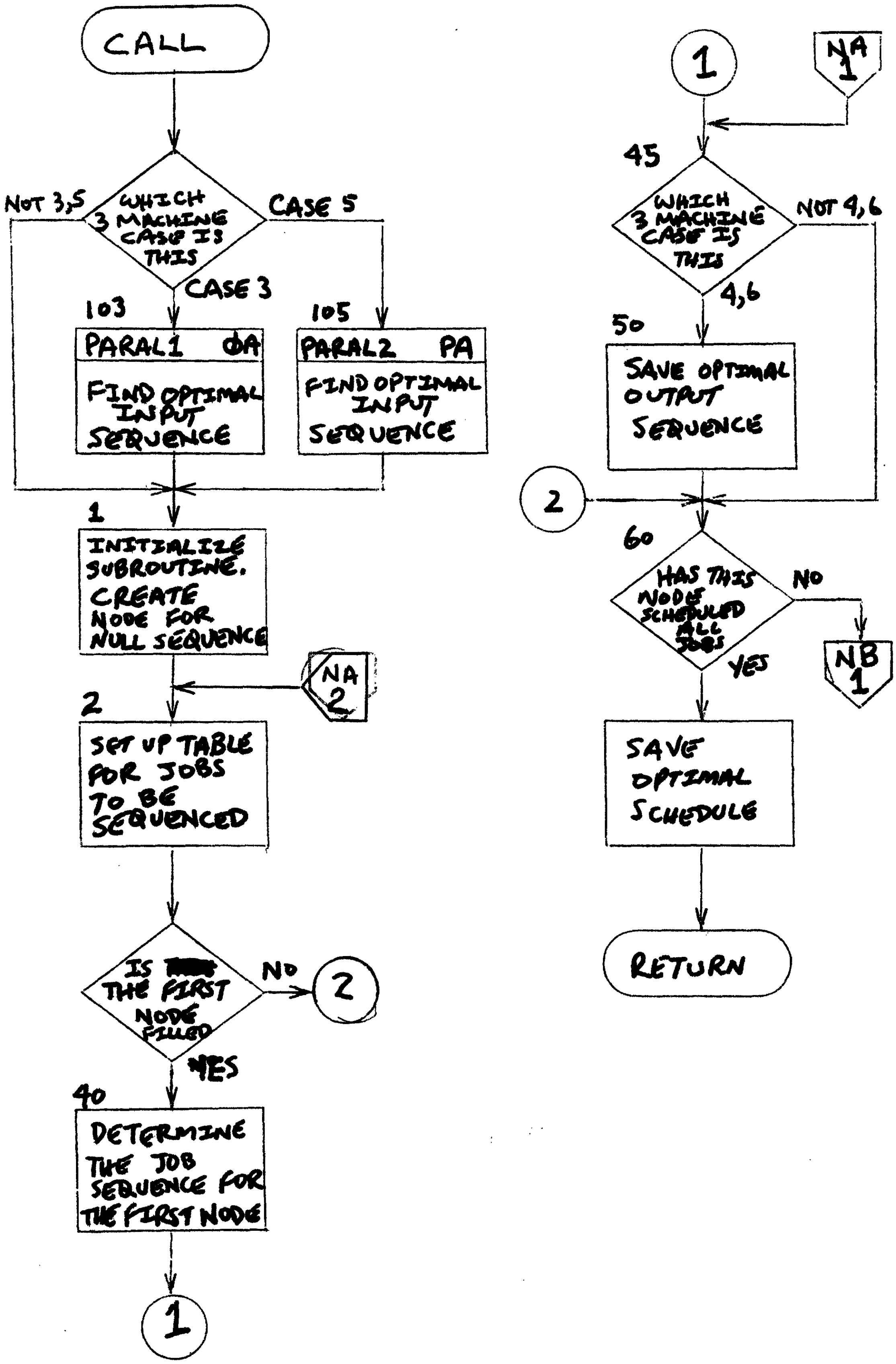
Subprogram SWITCH Chart LA



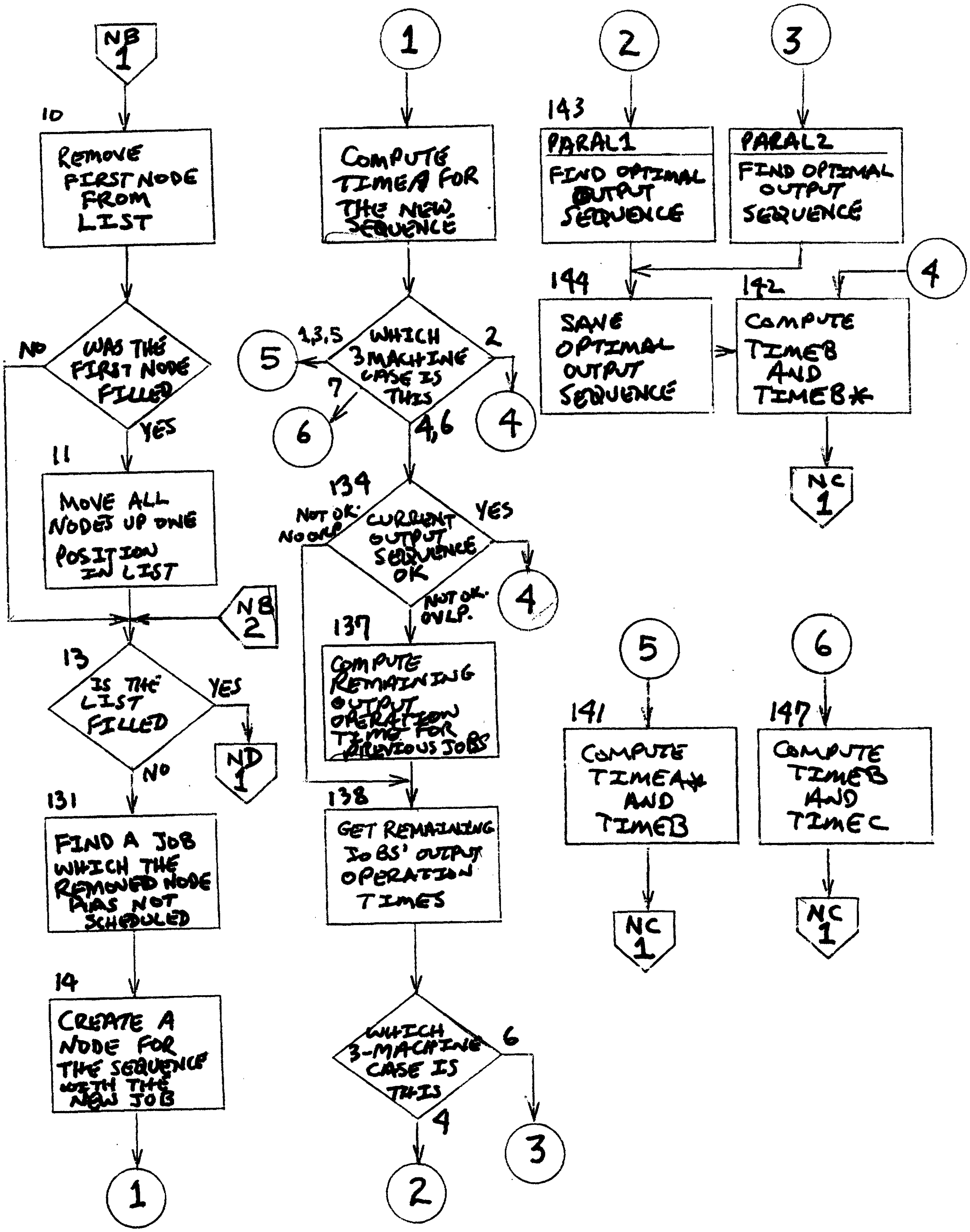
Subprogram JOHNSN Chart MA



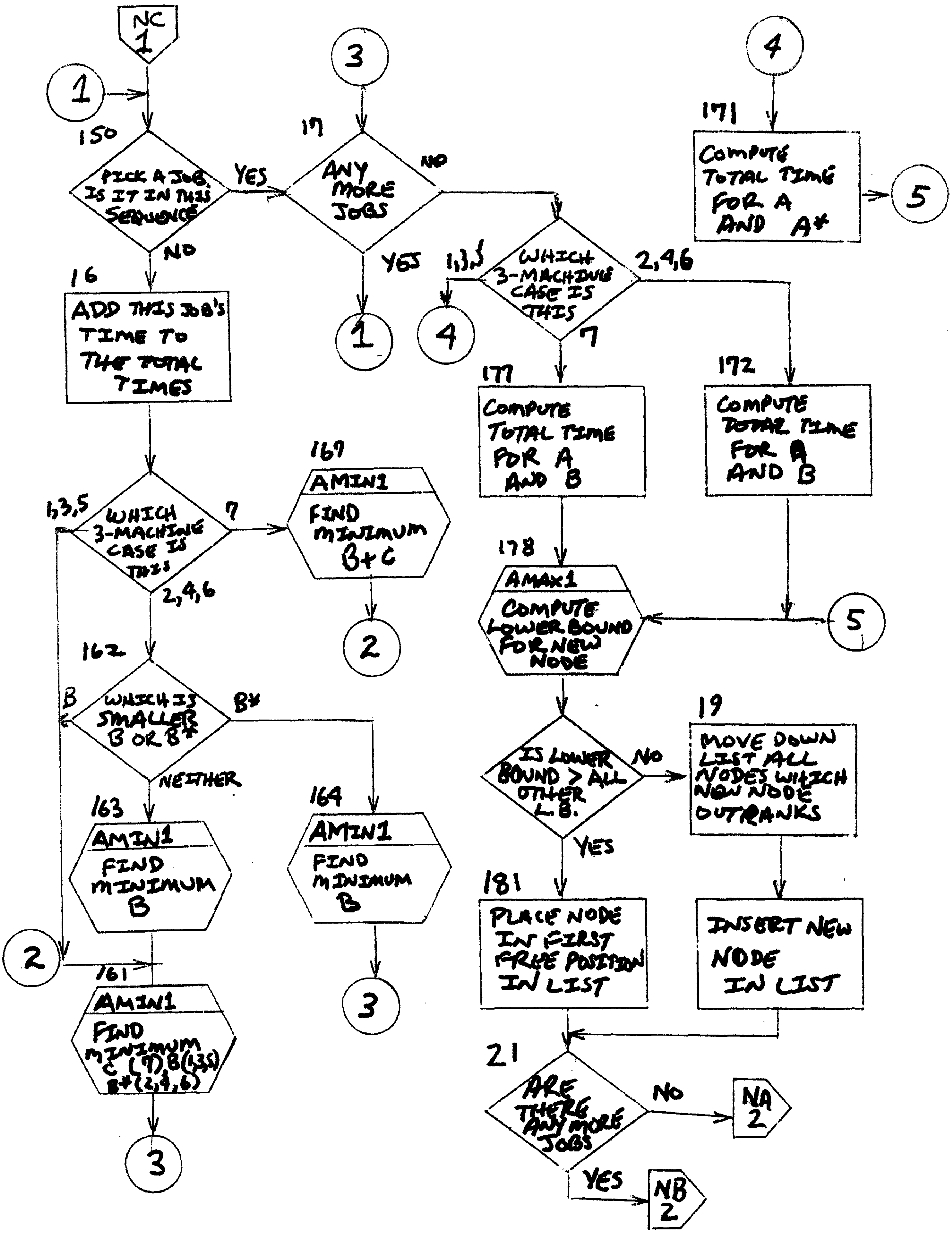
Subprogram IGNSCH Chart NA



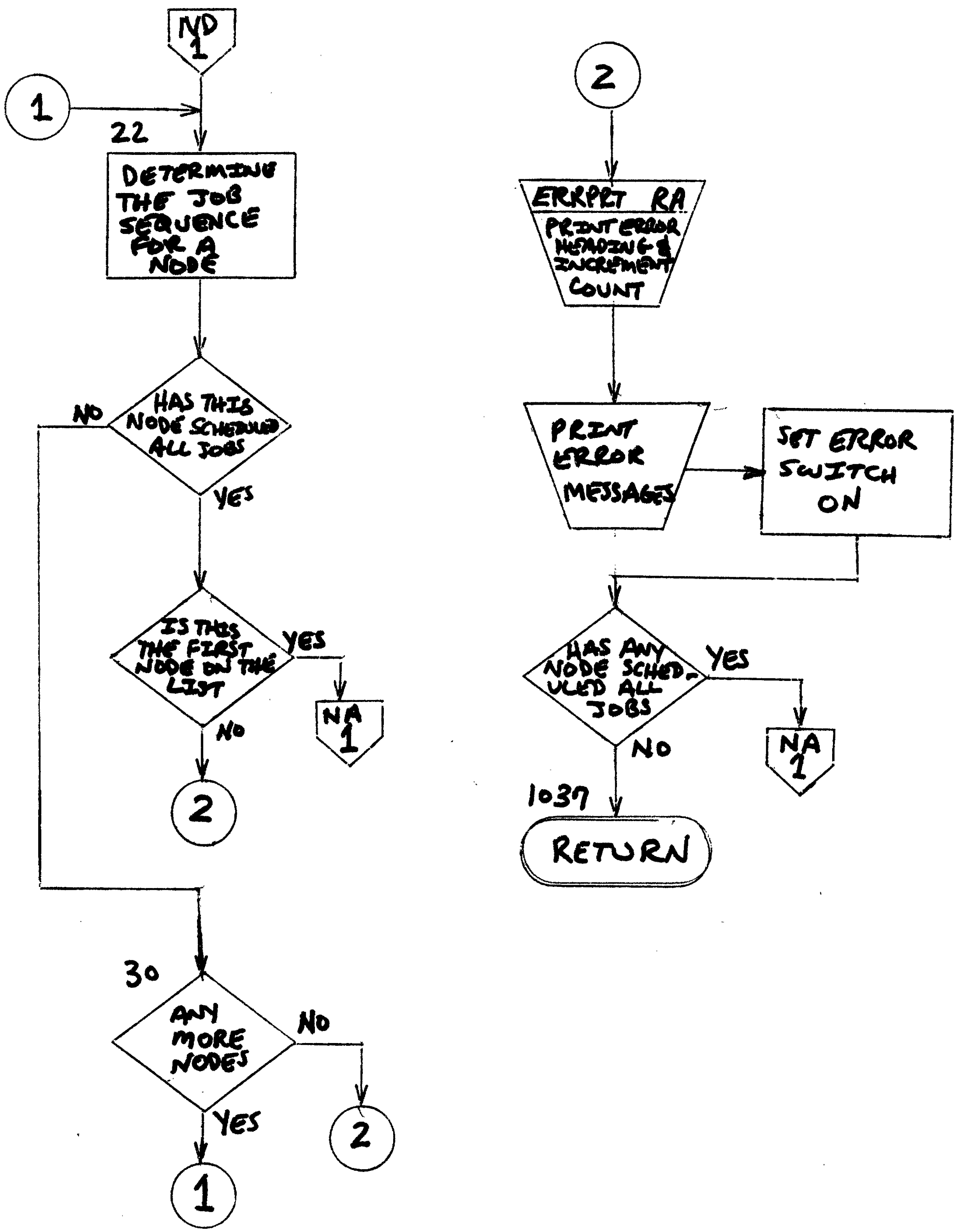
Subprogram IGNSCH Chart NB



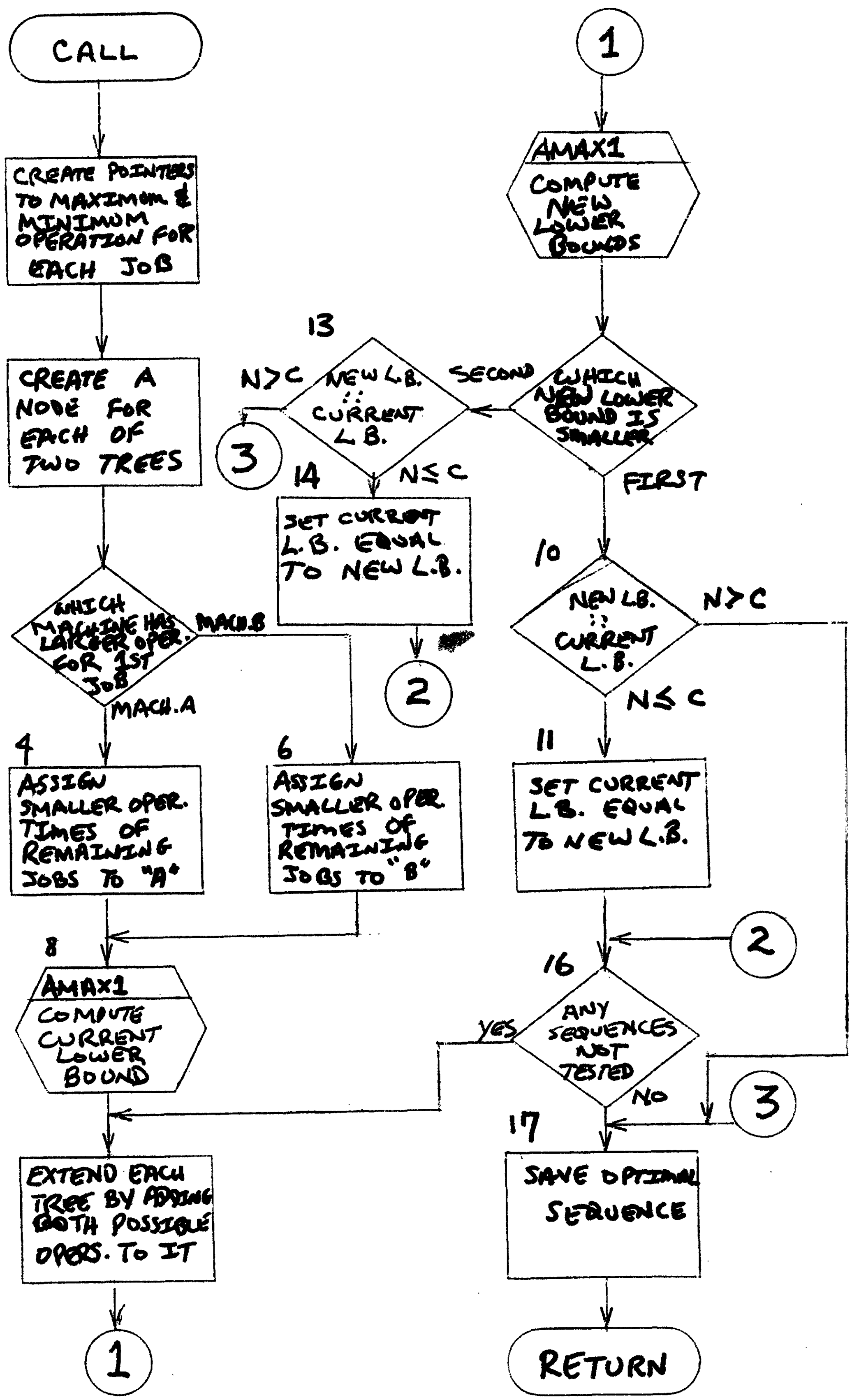
Subprogram IGNSCH Chart NC



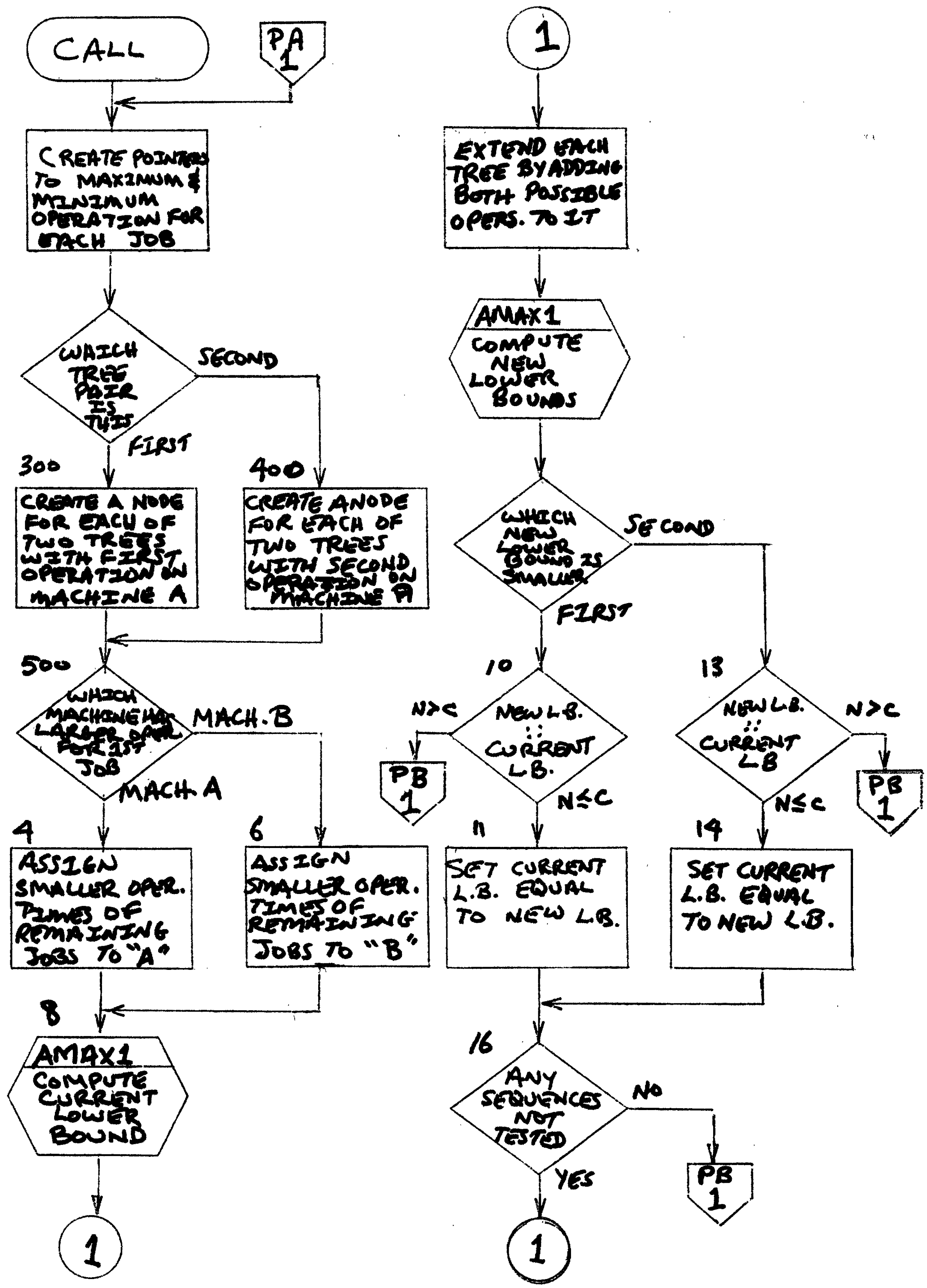
Subprogram IGNSCH   Chart ND



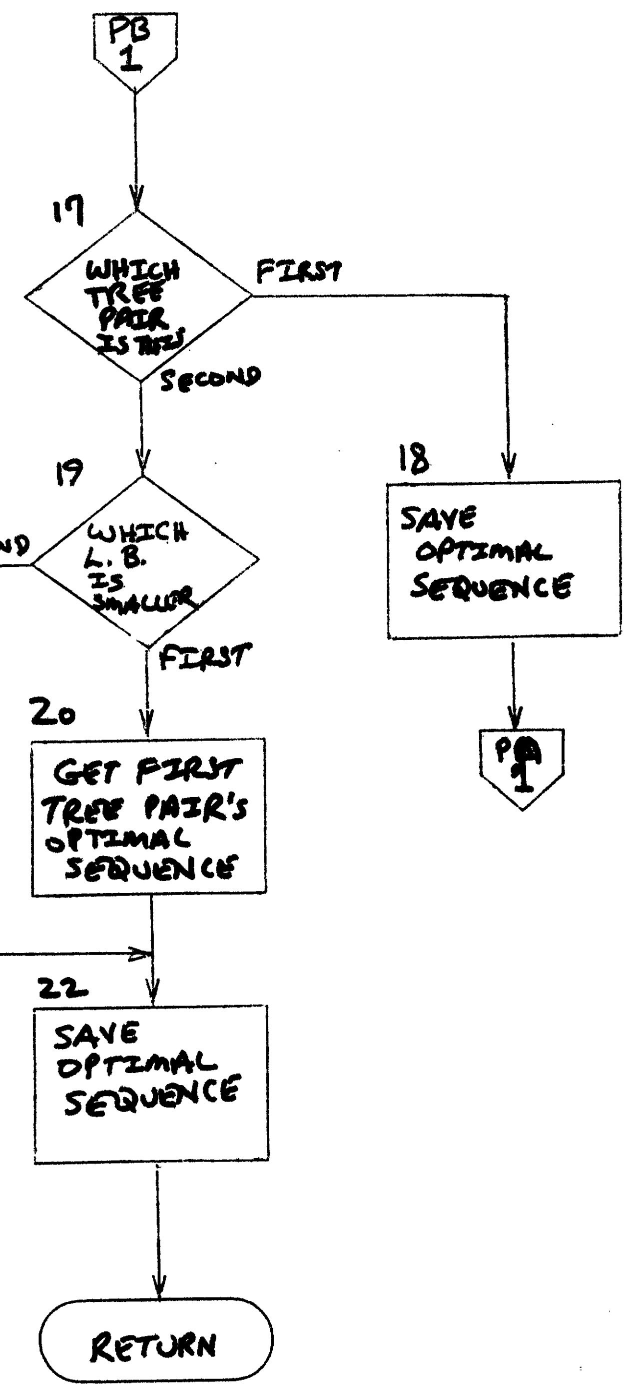
Subprogram PARAL1 Chart OA



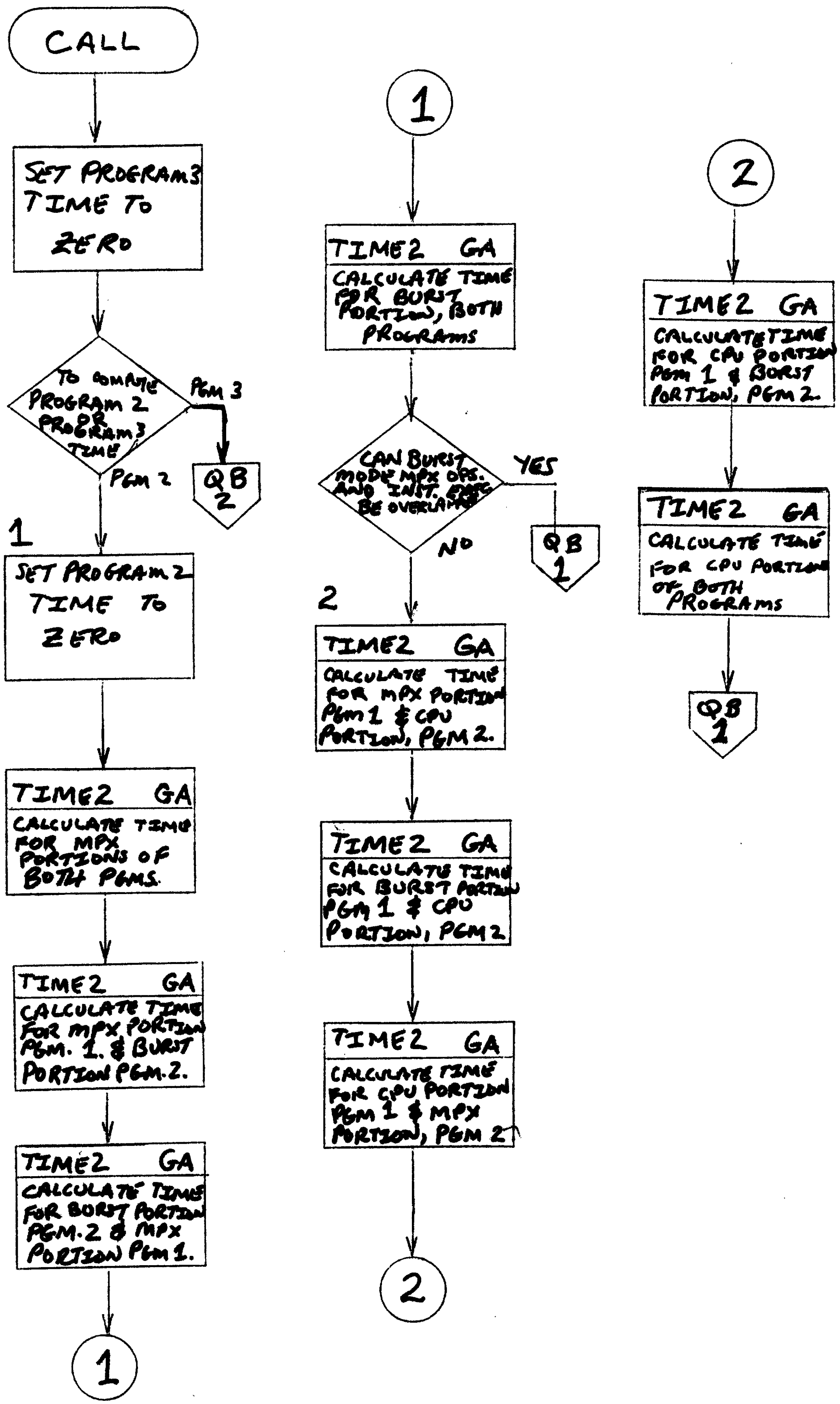
Subprogram PARAL2 Chart PA



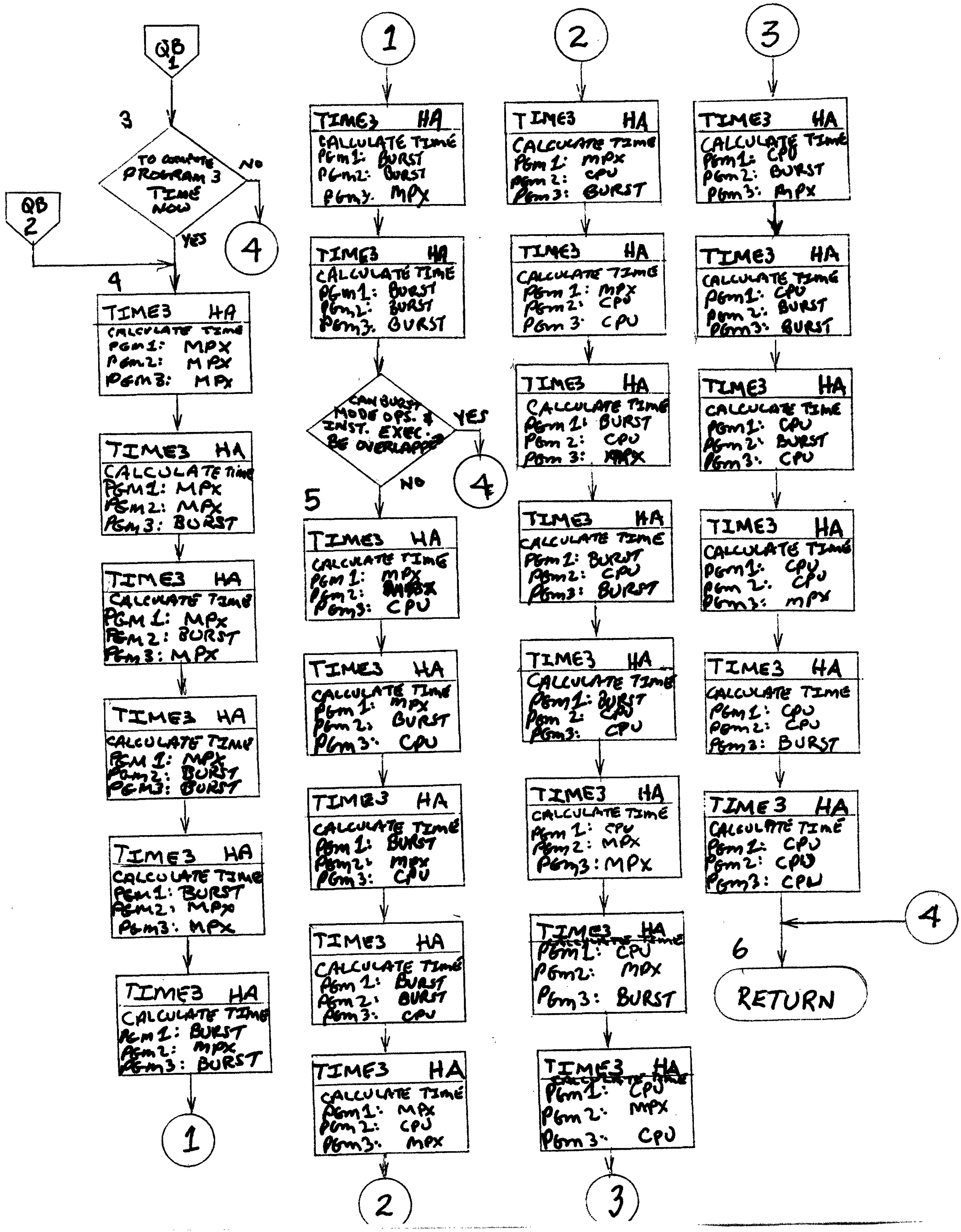
Subprogram PARAL2 Chart PB



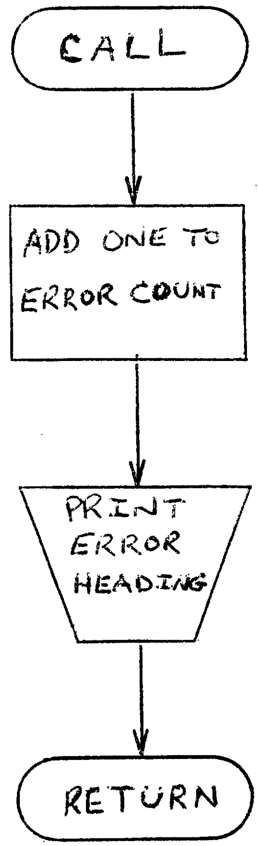
Subprogram TIME23 Chart QA



Subprogram TIME23 Chart QB



Subprogram ERRPRT Chart RA



FØRTRAN Source Statements

The following pages contain the source statements for the scheduling program. The program is written in the IBM System/360 Basic FORTRAN IV Language and has been tested on a System/360 Model I65 using the Operating System/360 (Release 11) FORTRAN G compiler. The input and output data set reference numbers are 5 and 6, respectively.

Room 14-0551
77 Massachusetts Avenue
Cambridge, MA 02139
Ph: 617.253.5668 Fax: 617.253.1690
Email: docs@mit.edu
http://libraries.mit.edu/docs

DISCLAIMER OF QUALITY

Due to the condition of the original material, there are unavoidable flaws in this reproduction. We have made every effort possible to provide you with the best copy available. If you are dissatisfied with this product and find it unusable, please contact Document Services as soon as possible.

Thank you.

The following pages contain text that are illegible.

```
C ******************************************************************************
C
C MAIN PROGRAM
C
C ******************************************************************************
      MDEV=10
      MCUU=20
      MFILE=20
      MJOB=20
      MDJOB=15
      MIJOB=10
      REAL I,J,G,NOVLP,LINCY,MULT
      INTEGER CPU,QE(5),ON(5),OFIXIT,CASE,FGUPPT(2),FGUIPT(2)
      INTEGER CUOPNT(20),FGCPPT(2),FGCIPT(2),FILPNT(20),ERROR,ERRORX
      INTEGER DEVICE,HHON(7),HHOFF(7),DON(20),HON(20),DOFF(20)
      INTEGER HOFF(20),TCASE,CIPT,CPPT,COUT,COUP,WEEK
      DIMENSION DEVTYP(10),DEVMOD(10),DEVCLS(10),DEVSPD(10),DEVSCP(10)
      DIMENSION DEVSTP(10),DEVLSS(10),DEVSRW(10),DEVOEN(10),DEVINT(10)
      DIMENSION CUU(20),FGCUUP(2),FGCUUT(2),TOTAL(3,20,2)
      DIMENSION IWORKH(7),IWORKM(7),IWAITH(7),IWAITM(7),RXFRCM(12,2)
      DIMENSION RXFRCB(12,2),RXFRCC(12,2),ISCHED(20,2),SCHED(3,20,4)
      DIMENSION FG(2),MMON(7),MMOFF(7),PRTCUU(30)
      DIMENSION MON(20),MOFF(20),TCPU(3,20,2),TNOVLP(3,20,2)
      DIMENSION FLEGPT(20,2),FLNMPT(20,2),FLECTR(2)
      DIMENSION FLIOPT(2),FLDEV(2),FLMULT(2),FLVOL(2),FDEV(3,20,20)
      DIMENSION SDEV(3,20,20),TMPXBA(3,20,2),TMPXBW(3,20,2)
      DIMENSION TSEL1A(3,20,2),TSEL1W(3,20,2),TSEL2A(3,20,2)
      DIMENSION TSEL2W(3,20,2),TSEL3A(3,20,2),TSEL3W(3,20,2)
      DIMENSION TSEL4A(3,20,2),TSEL4W(3,20,2),TSEL5A(3,20,2)
      DIMENSION TSEL5W(3,20,2),TSEL6A(3,20,2),TSEL6W(3,20,2)
      DIMENSION TSETUP(3,20,2),TWNDUP(3,20,2),TMPXMA(3,20,2)
      DIMENSION TMPXMW(3,20,2),FINSTX(2),FBLKFC(2),FLINCY(2),FSEEK(2)
      DOUBLE PRECISION BLANK8,PRECED,JOBNAM(20),NAME,PROGNM(2)
      DOUBLE PRECISION FILNAM(20),STEPNM(20),FILSEQ(20,2)
      ERROR=0
      ERRORX=0
      DO 9997 I1=1,3
      DO 9997 J1=1,MJOB
      DO 9997 K1=1,2
      TCPU(I1,J1,K1)=0.
      TNOVLP(I1,J1,K1)=0.
      TMPXBA(I1,J1,K1)=0.
      TMPXBW(I1,J1,K1)=0.
      TSEL1A(I1,J1,K1)=0.
      TSEL1W(I1,J1,K1)=0.
      TSEL2A(I1,J1,K1)=0.
      TSEL2W(I1,J1,K1)=0.
      TSEL3A(I1,J1,K1)=0.
```

```
0044             TSEL3W(I1,J1,K1)=0.                                                M0000046
0045             TSEL4A(I1,J1,K1)=0.                                                M0000047
0046             TSEL4W(I1,J1,K1)=0.                                                M0000048
0047             TSEL5A(I1,J1,K1)=0.                                                M0000049
0048             TSEL5W(I1,J1,K1)=0.                                                M0000050
0049             TSEL6A(I1,J1,K1)=0.                                                M0000051
0050             TSEL6W(I1,J1,K1)=0.                                                M0000052
0051             TSETUP(I1,J1,K1)=0.                                                M0000053
0052             TWNDUP(I1,J1,K1)=0.                                                M0000054
0053             TMPXMA(I1,J1,K1)=0.                                                M0000055
0054             TMPXMW(I1,J1,K1)=0.                                                M0000056
0055        9997 TOTAL(I1,J1,K1)=0.                                                 M0000057
0056             DO 9998 I1=1,3                                                     M0000058
0057             DO 9998 J1=1,MJOB                                                  M0000059
0058             DO 9998 K1=1,MCUU                                                  M0000060
0059             TDEV(I1,J1,K1)=0.                                                  M0000061
0060        9998 SDEV(I1,J1,K1)=0.                                                  M0000062
0061             DIMENSION FRCMF1(12,2),FRCBF1(12,2),FRCCF1(12,2),FRCMF2(12,2)      M0000070
0062             DIMENSION FRCBF2(12,2),FRCCF2(12,2),BGFRCM(12,2),BGFRCB(12,2)      M0000071
0063             DIMENSION BGFRCC(12,2),F1FRCM(12,2),F1FRCB(12,2),F1FRCC(12,2)      M0000072
0064             DIMENSION F2FRCM(12,2),F2FRCB(12,2),F2FRCC(12,2),FRACM(20,12,2)    M0000073
0065             DIMENSION FRACB(20,12,2),FRACC(20,12,2),FRCMMF(11,2),FRCMBF(11,2)  M0000074
0066             DIMENSION FRCMCF(11,2),FRCBMF(11,2),FRCBBF(11,2),FRCBCF(11,2)      M0000075
0067             DIMENSION FRCCMF(11,2),FRCCBF(11,2),FRCCCF(11,2),FRCMMX(11,2)      M0000076
0068             DIMENSION FRCMBX(11,2),FRCMCX(11,2),FRCBMX(11,2),FRCBBX(11,2)      M0000077
0069             DIMENSION FRCBCX(11,2),FRCCMX(11,2),FRCCBX(11,2),FRCCCX(11,2)      M0000078
0070             DIMENSION FRACMX(20,12,2),FRACBX(20,12,2),FRACCX(20,12,2)          M0000079
0071             DIMENSION QA(5),QB(5),QC(5),QD(5),QDX(5),QDP(5),QDPX(5),JCASE(20)  M0000081
           C
           C PRINT HEADING AND PROGRAM RESTRICTIONS
           C
0072             WRITE (6,4004)                                                     M0000100
0073        4004 FORMAT(1H1,46X,41HIBM SYSTEM/360 DOS/TOS SCHEDULING PROGRAM)       M0000101
0074             WRITE (6,4005)                                                     M0000109
0075        4005 FORMAT(1H0,56X,20HPROGRAM RESTRICTIONS)                            M0000110
0076             WRITE (6,4006) MDEV                                                M0000113
0077        4006 FORMAT(1H0,23X,75HMAXIMUM NUMBER OF DEVICE TYPES (DEFINED BY HD STM0000114
                1ATEMENTS) IN INSTALLATION =,I3)                                    M0000115
0078             WRITE (6,4007) MCUU                                                M0000118
0079        4007 FORMAT(1H0,23X,75HMAXIMUM NUMBER OF UNIT NUMBERS (DEFINED BY ID STM0000117
                1ATEMENTS) IN INSTALLATION =,I3)                                    M0000120
0080             WRITE (6,4008) MFILE                                               M0000123
0081        4008 FORMAT(1H0,23X,83HMAXIMUM NUMBER OF FILES (DEFINED BY JBD&JFD AND M0000124
                1JBIO&JFIO STATEMENTS) PER PROGRAM =,I3)                            M0000125
0082             J1=MJOB-MDJOB                                                      M0000128
0083             WRITE (6,4009) J1                                                  M0000129
0084        4009 FORMAT(1H0,23X,53HMAXIMUM NUMBER OF FULL WORKING DAY PROGRAMS PER M0000130
                1DAY =,I3)                                                          M0000131
```

```
0085              J1=MOJOB-MWJOB                                                      MO000134
0086              WRITE (6,4010) J1                                                   MO000135
0087         4010 FORMAT(1H0,23X,56HMAXIMUM NUMBER OF PARTIAL WORKING DAY PROGRAMS PMO000136
                 1ER DAY =,I3)
0088              WRITE (6,4011) MWJOB                                                MO000137
0089         4011 FORMAT(1H0,23X,43HMAXIMUM NUMBER OF WEEKLY PROGRAMS PER RUN =,I3) MO000140
0090              WRITE (6,4012)                                                      MO000147
0091         4012 FORMAT(113H0THE ABOVE MAXIMA MAY BE MODIFIED BY CHANGING THEIR DEFMO000148
                 1INITIONS IN THE SOURCE PROGRAM FOR THE SCHEDULING PROGRAM.)        MO000149
0092              WRITE (6,4013)                                                      MO000151
0093         4013 FORMAT(81H0FOR FURTHER INFORMATION SEE THE PROGRAM LOGIC MANUAL FOMO000151
                 1R THE SCHEDULING PROGRAM.)                                          MO000152
            C
            C READ DICTIONARY STATEMENT
            C
0094              READ (5,1) A,B,C,D,F,H,I,J,O,P,R,S,T,U,X,Z,D0,D1,D2,D3,D4,D5,D6,    MO000199
                 1D7,BLANK4,BLANK8                                                    MO000200
0095            1 FORMAT(24A1,A4,A8)                                                  MO000300
            C
            C READ HC STATEMENT
            C
0096              READ (5,2) A1,A2,CPU,AVINST,AVSEL,AVMPX                             MO000400
0097            2 FORMAT(2A1,I4,F3.1,F3.2,F3.1)                                       MO000500
0098              IF(A1-H)1002,3,1002                                                 MO000600
0099            3 IF(A2-C)1003,330,1003                                               MO000700
            C
            C CHECK FOR VALID CPU MODEL NUMBER
            C
0100          330 IF(CPU-2030)1330,4,340                                              MO000710
0101          340 IF(CPU-2040)1330,4,350                                              MO000720
0102          350 IF(CPU-2050)1330,4,365                                              MO000730
0103          365 IF(CPU-2065)1330,4,375                                              MO000740
0104          375 IF(CPU-2075)1330,4,1330                                             MO000750
            C
            C READ FIRST HD STATEMENT
            C
0105            4 READ (5,5) A1,A2,DEVTYP(1),DEVMOD(1),DEVCLS(1),DEVRCD(1),           MO000800
                 1DEVSPD(1),DEVSTP(1),DEVLSS(1),DEVSRW(1),DEVDEN(1),DEVINT(1)         MO000900
0106            5 FORMAT(2A1,A4,A2,A1,F3.0,F4.0,F2.1,3F3.0,F3.1)                      MO001000
0107              IF(A1-H)1005,6,1005                                                 MO001100
0108            6 IF(A2-D)1006,7,1006                                                 MO001200
            C
            C CHECK FOR VALID DEVICE CLASS
            C
0109            7 IF(DEVCLS(1)-D)8,10,8                                               MO001300
0110            8 IF(DEVCLS(1)-T)9,10,9                                               MO001400
0111            9 IF(DEVCLS(1)-U)1009,11,1009                                         MO001500
            C
```

```
              C COMPUTE DEVICE SPEED FOR NON-UNIT RECORD DEVICES
              C
 0112             10 DEVSPD(I)=(10000*DEVRCD(I))+DEVSPD(I)                          M0001600
 0113             11 IF(DEVINT(I)-0)1010,100,110                                    M0001650
              C
              C SAVE MULTIPLEXOR CHANNEL INTEFERENCE RATE
              C
 0114            100 DEVINT(I)=AVMPX                                                M0001700
 0115            110 IDEV=2                                                         M0001750
 0116             12 DO 21 I1=IDEV,MDEV                                             M0001800
              C
              C READ ADDITIONAL HD STATEMENTS AND/OR HZ STATEMENT
              C
 0117                READ (5,5) A1,A2,DEVTYP(I1),DEVMOD(I1),DEVCLS(I1),DEVRCD(I1),  M0001900
                    1DEVSPD(I1),DEVSTP(I1),DEVLSS(I1),DEVSRW(I1),DEVDEN(I1),DEVINT(I1) M0002000
 0118                IF(A1-H)1012,13,1012                                           M0002100
 0119             13 IF(A2-D)14,15,14                                               M0002200
 0120             14 IF(A2-Z)1014,22,1014                                           M0002300
              C
              C CHECK FOR VALID DEVICE CLASS
              C
 0121             15 IF(DEVCLS(I1)-D)16,18,16                                       M0002400
 0122             16 IF(DEVCLS(I1)-T)17,18,17                                       M0002500
 0123             17 IF(DEVCLS(I1)-U)1017,19,1017                                   M0002600
              C
              C COMPUTE DEVICE SPEED FOR NON-UNIT RECORD DEVICES
              C
 0124             18 DEVSPD(I1)=(10000*DEVRCD(I1))+DEVSPD(I1)                       M0002700
 0125             19 IF(DEVINT(I1)-0)1019,180,190                                   M0002750
              C
              C SAVE MULTIPLEXOR CHANNEL INTEFERENCE RATE
              C
 0126            180 DEVINT(I1)=AVMPX                                               M0002800
 0127            190 K=I1-1                                                         M0002850
              C
              C CHECK FOR DUPLICATE DEVICE TYPE AND MODEL NUMBER
              C
 0128                DO 21 J1=1,K                                                   M0002900
 0129                IF(DEVTYP(I1)-DEVTYP(J1))21,20,21                              M0003000
 0130             20 IF(DEVMOD(I1)-DEVMOD(J1))21,1020,21                            M0003100
 0131             21 CONTINUE                                                       M0003200
 0132                I1=I1+1                                                        M0003250
              C
              C NUMBER OF VALID HD STATEMENTS EQUALS MAXIMUM NUMBER OF DEVICES
              C
              C READ HZ STATEMENT
              C
 0133            210 READ (5,211) A1,A2                                             M0003300
```

```
0134         211 FORMAT(2A1)                                                        M0003400
0135             IF(A1-H)1211,212,1211                                              M0003500
0136         212 IF(A2-Z)1212,22,1212                                               M0003600
           C
           C SAVE NUMBER OF DEVICES
           C
0137          22 MDEV=I1-1                                                          M0003700
0138             I1=1                                                               M0003800
0139             J1=MIN0(23,MCUU)                                                   M0003900
           C
           C READ AN ID STATEMENT
           C
0140          23 READ (5,24) A1,A2,A3,A4,N1,(PRTCUU(K),K=1,23),A5                   M0004000
0141          24 FORMAT(2A1,A4,A2,I2,23A3,A1)                                       M0004100
0142             IF(A1-I)1024,240,1024                                              M0004200
0143         240 IF(A2-D)1240,241,1240                                              M0004300
           C
           C CHECK FOR VALID NUMBER OF UNIT DEFINITIONS ON STATEMENT
           C
0144         241 IF(MCUU-I1-N1+1)1241,25,25                                         M0004400
0145          25 IF(23-N1)1025,26,26                                                M0004500
0146          26 IF(N1-1)1026,260,260                                               M0004550
0147         260 DO 261 K=I1,J1                                                     M0004560
0148         261 CUU(K)=PRTCUU(K-I1+1)                                              M0004570
           C
           C RELATE UNIT TO DEVICE TYPE AND MODEL NUMBER
           C
0149             DO 28 L=1,MDEV                                                     M0004600
0150             IF(A3-DEVTYP(L))28,27,28                                           M0004700
0151          27 IF(A4-DEVMOD(L))28,280,28                                          M0004800
0152          28 CONTINUE                                                           M0004900
0153             GO TO 1028                                                         M0005000
0154         280 M=I1+N1-1                                                          M0005100
           C
           C CHECK FOR DUPLICATE UNIT NUMBER
           C
0155             DO 30 K=I1,M                                                       M0005200
0156             GO TO (302,29),I1                                                  M0005250
0157          29 N=K-1                                                              M0005300
0158             DO 300 NN=1,N                                                      M0005400
0159             IF(CUU(K)-CUU(NN))300,1029,300                                     M0005500
0160         300 CONTINUE                                                           M0005600
           C
           C CHECK FOR VALID CHANNEL NUMBER
           C
0161         302 IF(ABS(CUU(K))-ABS(D0))1300,1300,301                               M0005700
0162         301 IF(ABS(D7)-ABS(CUU(K)))1300,1300,30                                M0005800
           C
```

```
              C SAVE POINTER TO DEVICE TYPE AND MODEL NUMBER
              C
0163             30 CUUPNT(K)=L                                                   M0005900
              C
              C CHECK FOR ADDITIONAL ID STATEMENTS
              C
0164            303 IF(A5-BLANK4)31,32,31                                         M0006000
0165             31 I1=I1+N1                                                      M0006100
0166                IF(I1-MCUU)310,310,32                                         M0006150
0167            310 J1=MIN0(I1+22,MCUU)                                           M0006200
0168                GO TO 23                                                      M0006300
              C
              C SAVE NUMBER OF UNITS
              C
0169             32 MCUU=K                                                        M0006400
0170                I2=1                                                          M0006410
0171            320 DO 73 IFG=I2,2                                                M0006430
              C
              C READ AN IF STATEMENT
              C
0172                READ (5,33) A1,A2,N1,A3,A4,PROGNM(IFG),FGCUUP(IFG),FGCUUI(IFG), M0006500
                   1FINSTX(IFG),FBLKFC(IFG),FLTNCY(IFG),FSEEK(IFG)                M0006600
0173             33 FORMAT(2A1,I1,2A1,A8,2A3,F3.0,F2.0,2F3.0)                     M0006800
0174                IF(A1-I)1033,34,1033                                          M0007000
0175             34 IF(A2-F)1034,35,1034                                          M0007100
0176             35 IF(N1-IFG)1035,36,1035                                        M0007200
              C
              C CHECK FOR VALID FOREGROUND PROGRAM TYPE
              C
0177             36 IF(A3-BLANK4)38,37,38                                         M0007300
0178             37 IF(A4-BLANK4)1037,63,1037                                     M0007400
0179             38 IF(A3-P)41,39,41                                              M0007500
0180             39 IF(A4-BLANK4)1037,40,1037                                     M0007600
              C
              C FOREGROUND PROGRAM IS A PRINTER TRANSCRIPTION PROGRAM
              C
0181             40 FG(IFG)=P                                                     M0007700
0182                GO TO 49                                                      M0007800
0183             41 IF(A3-R)44,42,44                                              M0007900
0184             42 IF(A4-BLANK4)1037,43,1037                                     M0008000
              C
              C FOREGROUND PROGRAM IS A READER TRANSCRIPTION PROGRAM
              C
0185             43 FG(IFG)=R                                                     M0008100
0186                GO TO 49                                                      M0008200
0187             44 IF(A3-T)1044,45,1044                                          M0008300
0188             45 IF(A4-T)46,47,46                                              M0008400
0189             46 IF(A4-O)1046,48,1046                                          M0008500
```

```
              C
              C FOREGROUND PROGRAM IS A TELECOMMUNICATIONS INPUT TRANSCRIPTION PROGRAM
              C
0190             47 FG(IFG)=I                                                          M0008600
0191                GO TO 49                                                           M0008700
              C
              C FOREGROUND PROGRAM IS A TELECOMMUNICATIONS OUTPUT TRANSCRIPTION PROGRAM
              C
0192             48 FG(IFG)=0                                                          M0008800
              C
              C CHECK FOR EXISTENCE OF PROGRAM NAME
              C
0193             49 IF(PROGNM(IFG)-BLANK8)490,1049,490                                 M0008900
              C
              C CHECK FOR VALID UNIT NUMBER
              C
0194            490 DO 51 I1=1,MCUU                                                    M0008950
0195                IF(FGCUUP(IFG)-CUU(I1))51,50,51                                    M0009000
0196             50 J1=CUUPNT(I1)                                                      M0009100
0197                FGCPPT(IFG)=J1                                                     M0009200
0198                FGUPPT(IFG)=I1                                                     M0009250
              C
              C VERIFY THAT PERIPHERAL DEVICE IS A UNIT RECORD DEVICE
              C
0199                IF(DEVCLS(J1)-U)1050,52,1050                                       M0009300
0200             51 CONTINUE                                                           M0009400
0201                GO TO 1051                                                         M0009500
              C
              C CHECK FOR VALID UNIT NUMPER
              C
0202             52 DO 55 I1=1,MCUU                                                    M0009600
0203                IF(FGCUUI(IFG)-CUU(I1))55,53,55                                    M0009700
0204             53 J1=CUUPNT(I1)                                                      M0009800
0205                FGCIPT(IFG)=J1                                                     M0009900
0206                FGUIPT(IFG)=I1                                                     M0009950
              C
              C VERIFY THAT INTERMEDIATE STORAGE DEVICE IS A DISK OR TAPE DEVICE
              C
0207                IF(DEVCLS(J1)-D)54,56,54                                           M0010000
0208             54 IF(DEVCLS(J1)-T)1054,73,1054                                       M0010100
0209             55 CONTINUE                                                           M0010200
0210                GO TO 1055                                                         M0010300
0211             56 IF(FLTNCY(IFG)-0.)1056,57,58                                       M0010400
              C
              C SAVE LATENCY TIME FOR DISK DEVICE
              C
0212             57 FLTNCY(IFG)=DEVLSS(J1)                                             M0010500
0213             58 IF(FSEEK(IFG)-0.)1058,59,73                                        M0010600
```

```
              C
              C SAVE SEEK TIME FOR DISK DEVICE
              C
  0214          59 SSEEK(IFG)=DEVSRW(J1)                                      MOO10700
  0215             GO TO 73                                                   MOO12400
              C
              C THERE IS NO PROGRAM ASSIGNED TO THIS FOREGROUND AREA
              C
  0216          63 FG(IFG)=BLANK4                                             MOO12600
  0217          73 CONTINUE                                                   MOO12700
              C
              C CHECK FOR DEVICE CONFLICT
              C
  0218             IF(FG(1)-BLANK4)64,680,64                                  MOO12800
  0219          64 IF(FG(2)-BLANK4)65,680,65                                  MOO12900
  0220          65 IF(FGCUUP(1)-FGCUUP(2))67,66,67                            MOO13000
  0221          66 A0=FGCUUP(1)                                               MOO13100
  0222             GO TO 1066                                                 MOO13200
  0223          67 IF(FGCUUI(1)-FGCUUI(2))680,68,680                          MOO13300
  0224          68 A0=FGCUUI(1)                                               MOO13400
  0225             J1=FGCIPT(1)                                               MOO13500
  0226             IF(DEVCLS(J1)-T)680,1065,680                               MOO13600
              C
              C COMPUTE RESOURCE USE FRACTIONS FOR EACH NORMAL FOREGROUND ASSIGNMENT
              C
  0227         680 DO 689 I1=1,2                                              MOO13700
              C
              C IS THERE A PROGRAM ASSIGNED TO THIS FOREGROUND AREA
              C
  0228             IF(FG(I1)-BLANK4)681,689,681                               MOO13710
              C
              C YES.  CALCULATE DEVICE, CHANNEL, CPU, AND SET UP TIMES FOR UNIT RECORD DEVICE
              C
  0229         681 J1=FGCPPT(I1)                                              MOO13720
  0230             CALL UNITX (TDEV(1,1,1),TMPXMA(1,1,1),TSEL1A(1,1,1),TSEL2A(1,1,1),MOO13730
                  1TSEL3A(1,1,1),TSEL4A(1,1,1),TSEL5A(1,1,1),TSEL6A(1,1,1),     MOO13740
                  2TCPU(1,1,1),TSETUP(1,1,1),TNOVLP(1,1,1),DEVSPD(J1),DEVRCD(J1), MOO13750
                  3DEVSTP(J1),1.,X,1.,FINSTX(I1),AVINST,AVSEL,DEVINT(J1),0.,   MOO13760
                  4FGCUUP(I1),D1,D2,D3,D4,D5,D6,X)                            MOO13770
  0231             K1=FGCIPT(I1)                                              MOO13780
              C
              C IS THE INTERMEDIATE STORAGE DEVICE DISK OR TAPE
              C
  0232             IF(DEVCLS(K1)-D)682,683,682                                MOO13790
              C
              C TAPE.  CALCULATE DEVICE, CHANNEL, CPU, SET UP, AND REWIND TIMES
              C
  0233         682 CALL TAPEX (TDEV(1,1,2),TMPXBA(1,1,1),TSEL1A(1,1,1),TSEL2A(1,1,1),MOO13800
```

```
                 1TSEL3A(1,1,1),TSEL4A(1,1,1),TSEL5A(1,1,1),FSEL6A(1,1,1),          MO013810
                 2TCPU(1,1,1),TSETUP(1,1,1),TWNDUP(1,1,1),TNOVLP(1,1,1),DEVSPD(K1), MO013820
                 3DEVSTP(K1),1.,DEVRCD(J1)+1.,FBLKFC(I1),BLANK4,DEVLSS(K1),         MO013830
                 4DEVSRV(K1),DEVDEN(K1),0.,AVINST,AVSEL,DEVINT(K1),0.,FGCUIT(I1),I1,MO013840
                 5D2,D3,D4,D5,D6,X)                                                 MO013850
0234              GO TO 684                                                         MO013860
            C
            C DISK.  CALCULATE DEVICE, CHANNEL, CPU, AND SET UP TIMES
            C
0235          683 CALL DISKX (TDEV(1,1,2),TMPXBA(1,1,1),TMPXBW(1,1,1),TSEL1A(1,1,1),MO013870
                 1TSEL1W(1,1,1),TSEL2A(1,1,1),TSEL2W(1,1,1),TSEL3A(1,1,1),          MO013880
                 2TSEL3W(1,1,1),TSEL4A(1,1,1),TSEL4W(1,1,1),TSEL5A(1,1,1),          MO013890
                 3TSEL5W(1,1,1),TSEL6A(1,1,1),TSEL6W(1,1,1),TCPU(1,1,1),            MO013900
                 4TSETUP(1,1,1),TNOVLP(1,1,1),DEVSPD(K1),DEVSTP(K1),1.,             MO013910
                 5DEVRCD(J1)+1.,FBLKFC(I1),BLANK4,FLTNCY(I1),FSEEK(I1),0.,AVINST,   MO013920
                 6AVSEL,DEVINT(K1),0.,FGCUIT(I1),D1,D2,D3,D4,D5,D6,X)               MO013930
            C
            C WHICH FOREGROUND PROGRAM IS THIS
            C
0236          684 GO TO (685,686),I1                                                MO013940
            C
            C FIRST.  COMPUTE RESOURCE USE FRACTIONS
            C
0237          685 CALL FRCTN (TOTAL(1,1,1),TCPU(1,1,1),TSEL1A(1,1,1),TSEL1W(1,1,1), MO013950
                 1TSEL2A(1,1,1),TSEL2W(1,1,1),TSEL3A(1,1,1),TSEL3W(1,1,1),          MO013960
                 2TSEL4A(1,1,1),TSEL4W(1,1,1),TSEL5A(1,1,1),TSEL5W(1,1,1),          MO013970
                 3TSEL6A(1,1,1),TSEL6W(1,1,1),TMPXMA(1,1,1),TMPXMW(1,1,1),          MO013980
                 4TMPXBA(1,1,1),TMPXBW(1,1,1),TNOVLP(1,1,1),CPU,FRCMF1,FRCBF1,      MO013990
                 5FRCCF1)                                                           MO014000
0238              GO TO 687                                                         MO014010
            C
            C SECOND.  COMPUTE RESOURCE USE FRACTIONS
            C
0239          686 CALL FRCTN (TOTAL(1,1,1),TCPU(1,1,1),TSEL1A(1,1,1),TSEL1W(1,1,1), MO014020
                 1TSEL2A(1,1,1),TSEL2W(1,1,1),TSEL3A(1,1,1),TSEL3W(1,1,1),          MO014030
                 2TSEL4A(1,1,1),TSEL4W(1,1,1),TSEL5A(1,1,1),TSEL5W(1,1,1),          MO014040
                 3TSEL6A(1,1,1),TSEL6W(1,1,1),TMPXMA(1,1,1),TMPXMW(1,1,1),          MO014050
                 4TMPXBA(1,1,1),TMPXBW(1,1,1),TNOVLP(1,1,1),CPU,FRCMF2,FRCBF2,      MO014060
                 5FRCCF2)                                                           MO014070
0240          687 TOTAL(1,1,1)=0.                                                   MO014080
0241              TDEV(1,1,1)=0.                                                    MO014090
0242              TDEV(1,1,2)=0.                                                    MO014100
0243              TCPU(1,1,1)=0.                                                    MO014110
0244              TSEL1A(1,1,1)=0.                                                  MO014120
0245              TSEL1W(1,1,1)=0.                                                  MO014130
0246              TSEL2A(1,1,1)=0.                                                  MO014140
0247              TSEL2W(1,1,1)=0.                                                  MO014150
0248              TSEL3A(1,1,1)=0.                                                  MO014160
```

```
0249          TSEL3W(1,1,1)=0.                                       MO014170
0250          TSEL4A(1,1,1)=0.                                       MO014180
0251          TSEL4W(1,1,1)=0.                                       MO014190
0252          TSEL5A(1,1,1)=0.                                       MO014200
0253          TSEL5W(1,1,1)=0.                                       MO014210
0254          TSEL6A(1,1,1)=0.                                       MO014220
0255          TSEL6W(1,1,1)=0.                                       MO014230
0256          TMPXMA(1,1,1)=0.                                       MO014240
0257          TMPXMW(1,1,1)=0.                                       MO014250
0258          TMPXBA(1,1,1)=0.                                       MO014260
0259          TMPXBW(1,1,1)=0.                                       MO014270
0260          TNOVLP(1,1,1)=0.                                       MO014280
0261          TSETUP(1,1,1)=0.                                       MO014290
0262          TENDUP(1,1,1)=0.                                       MO014300
0263      689 CONTINUE                                               MO014310
        C
        C DETERMINE WHICH SCHEDULING ALGORITHM TO USE FOR FULL WORKING DAY PROGRAMS
        C
0264          IF(FG(1)-BLANK4)691,690,691                            MO014400
0265      690 IF(FG(2)-BLANK4)694,720,694                            MO014410
0266      691 IF(FG(2)-BLANK4)696,692,696                            MO014420
0267      692 IF(FG(1)-R)693,728,693                                 MO014430
0268      693 IF(FG(1)-I)729,728,729                                 MO014440
0269      694 IF(FG(2)-R)695,728,695                                 MO014450
0270      695 IF(FG(2)-I)729,728,729                                 MO014460
        C
        C IS THE SAME TYPE OF PROGRAM ASSIGNED TO BOTH FOREGROUND AREAS
        C
0271      696 IF(FG(1)-FG(2))697,704,697                             MO014470
0272      697 IF(FG(1)-R)698,700,698                                 MO014480
0273      698 IF(FG(1)-I)699,701,699                                 MO014490
0274      699 IF(FG(1)-P)703,702,703                                 MO014500
0275      700 IF(FG(2)-I)727,721,727                                 MO014510
0276      701 IF(FG(2)-R)727,721,727                                 MO014520
0277      702 IF(FG(2)-D)727,722,727                                 MO014530
0278      703 IF(FG(2)-P)727,722,727                                 MO014540
        C
        C YES.  ARE THE PROGRAM CHARACTERISTICS IDENTICAL
        C
0279      704 IF(FGCPPT(1)-FGCPPT(2))713,705,713                     MO014550
0280      705 IF(FGCIPT(1)-FGCIPT(2))713,706,713                     MO014560
0281      706 IF(FINSTX(1)-FINSTX(2))713,707,713                     MO014570
0282      707 IF(FBLKFC(1)-FBLKFC(2))713,708,713                     MO014580
0283      708 J1=FGCIPT(1)                                           MO014590
0284          IF(DEVCLS(J1)-D)711,709,711                            MO014600
0285      709 IF(FLTNCY(1)-FLTNCY(2))713,710,713                     MO014610
0286      710 IF(FSEEK(1)-FSEEK(2))713,711,713                       MO014620
0287      711 IF(FG(1)-R)712,723,712                                 MO014630
```

```
0288          712 IF(FG(1)-I)724,723,724                                   M0014640
0289          713 IF(FG(1)-R)714,725,714                                   M0014650
0290          714 IF(FG(1)-I)726,725,726                                   M0014660
0291          721 CASE=1                                                   M0014670
0292              GO TO 715                                                M0014680
0293          722 CASE=2                                                   M0014690
0294              GO TO 715                                                M0014700
0295          723 CASE=3                                                   M0014710
0296              GO TO 715                                                M0014720
0297          724 CASE=4                                                   M0014730
0298              GO TO 715                                                M0014740
0299          725 CASE=5                                                   M0014750
0300              GO TO 715                                                M0014760
0301          726 CASE=6                                                   M0014770
0302              GO TO 715                                                M0014780
0303          727 CASE=7                                                   M0014790
        C
        C COMPUTE SECOND FOREGROUND AREA'S MULTIPROGRAMMING TIME
        C
0304          715 CALL TIME23 (FRCMF1,FRCBF1,FRCCF1,FRCMF2,FRCBF2,FRCCF2,FRCMF1,   M0014800
             1FRCBF1,FRCCF1,FRCMMF,FRCMBF,FRCMCF,FRCBMF,FRCBBF,FRCBCF,FRCCMF,    M0014810
             2FRCCBF,FRCCCF,1,CPU,TIMEF,TIMEB)                             M0014820
0305              GO TO 730                                                M0014830
0306          728 CASE=8                                                   M0014840
0307              GO TO 730                                                M0014850
0308          729 CASE=9                                                   M0014860
0309              GO TO 730                                                M0014870
0310          720 CASE=10                                                  M0014880
        C
        C READ IS STATEMENT
        C
0311          730 READ (5,74) A1,A2,N1,(HHON(II),MMON(II),HHOFF(II),MMOFF(II),     M0016000
             1II=1,7)                                                      M0016050
0312           74 FORMAT(2A1,I1,7(4I2))                                    M0016100
0313              IF(A1-I)1074,75,1074                                     M0016200
0314           75 IF(A2-S)1075,750,1075                                    M0016300
        C
        C CHECK FOR VALID NUMBER OF DAYS IN WEEK
        C
0315          750 IF(N1-7)751,751,1750                                     M0016330
0316          751 IF(N1-1)1751,752,752                                     M0016360
        C
        C SAVE NUMBER OF DAYS IN WEEK
        C
0317          752 WEEK=N1                                                  M0016390
        C
        C VERIFY DAILY BEGIN AND END TIMES
        C
```

```
      76 I1=1                                                           M0016400
     760 IF(HHON(I1)-23)770,770,1760                                    M0016500
     770 IF(HHON(I1)-0)1770,77,77                                       M0016550
      77 IF(HHOFF(I1)-23)780,780,1077                                   M0016600
     780 IF(HHOFF(I1)-0)1780,78,78                                      M0016650
      78 IF(MMON(I1)-59)790,790,1078                                    M0016700
     790 IF(MMON(I1)-0)1790,79,79                                       M0016750
      79 IF(MMOFF(I1)-59)800,800,1079                                   M0016800
     800 IF(MMOFF(I1)-0)1800,80,80                                      M0016850
      80 IF(HHOFF(I1)-HHON(I1))1081,81,82                               M0016900
      81 IF(MMOFF(I1)-MMON(I1))1081,1081,82                             M0017000
      82 I1=I1+1                                                        M0017100
         IF(I1-WEEK)760,760,850                                         M0017150
C
C READ I7 STATEMENT
C
     850 READ (5,85) A1,A2                                              M0017700
      85 FORMAT(2A1)                                                    M0017800
         IF(A1-I)1085,86,1085                                           M0017900
      86 IF(A2-7)1086,5015,1086                                         M0018000
C
C PRINT CPU INPUT SUMMARY
C
    5015 WRITE (6,4015)                                                 M0018001
    4015 FORMAT(1H1,51X,31HCENTRAL PROCESSING UNIT SUMMARY//)           M0018002
         WRITE (6,4016) CPU                                             M0018003
    4016 FORMAT(//18H CPU MODEL NUMBER=,29X,I4)                         M0018004
         WRITE (6,4017) AVINST                                          M0018005
    4017 FORMAT(27HOAVERAGE INSTRUCTION TIME =,20X,F5.1,1X,12HMICROSECONDS)M0018006
         WRITE (6,4018) AVSEL                                           M0018007
    4018 FORMAT(44HOAVERAGE SELECTOR CHANNEL INTERFERENCE TIME=,3X,F5.2,1X,M0018008
        121HMICROSECONDS PER BYTE)                                      M0018009
         WRITE (6,4019) AVMPX                                           M0018010
    4019 FORMAT(47HOAVERAGE MULTIPLEXOR CHANNEL INTERFERENCE TIME=,F5.1,1X,M0018011
        121HMICROSECONDS PER BYTE)                                      M0018012
C
C PRINT FOREGROUND AREA INPUT SUMMARY
C
         WRITE (6,4020)                                                 M0018013
    4020 FORMAT(/////1H ,49X,34HNORMAL FOREGROUND AREA ASSIGNMENTS//)   M0018014
         DO 5020 I1=1,2                                                 M0018015
         WRITE (6,4021) I1                                              M0018016
    4021 FORMAT(//16H FOREGROUND AREA,I2)                               M0018017
         IF(FG(I1)-BLANK4)5011,5022,5011                                M0018018
    5022 WRITE (6,4022)                                                 M0018019
    4022 FORMAT(44HONO FOREGROUND PROGRAM ASSIGNED TO THIS AREA)        M0018020
         GO TO 5020                                                     M0018021
    5011 IF(FG(I1)-R)5012,5023,5012                                     M0018022
```

```
0355       5012 IF(FG(I1)-P)5013,5024,5013                                     M0018023
0356       5013 IF(FG(I1)-I)5026,5025,5026                                     M0018024
0357       5023 WRITE (6,4023)                                                 M0018025
0358       4023 FORMAT(56HOCARD READER TRANSCRIPTION PROGRAM ASSIGNED TO THIS AREAM0018026
               1)                                                              M0018027
0359            GO TO 5027                                                     M0018028
0360       5024 WRITE (6,4024)                                                 M0018029
0361       4024 FORMAT(52HOPRINTER TRANSCRIPTION PROGRAM ASSIGNED TO THIS AREA)   M0018030
0362            GO TO 5027                                                     M0018031
0363       5025 WRITE (6,4025)                                                 M0018032
0364       4025 FORMAT(69HOTELECOMMUNICATIONS INPUT TRANSCRIPTION PROGRAM ASSIGNEDM0018033
               1 TO THIS AREA)                                                 M0018034
0365            GO TO 5027                                                     M0018035
0366       5026 WRITE (6,4026)                                                 M0018036
0367       4026 FORMAT(70HOTELECOMMUNICATIONS OUTPUT TRANSCRIPTION PROGRAM ASSIGNEM0018037
               1D TO THIS AREA)                                                M0018038
0368       5027 WRITE (6,4027) PROGNM(I1)                                      M0018039
0369       4027 FORMAT(16HOPROGRAM NAME IS,29X,A8)                             M0018040
0370            IFIXIT=FINSTX(I1)                                              M0018041
0371            WRITE (6,4028) IFIXIT                                          M0018042
0372       4028 FORMAT(44HONUMBER OF INSTRUCTIONS EXECUTED PER RECORD=,1X,I3)   M0018043
0373            WRITE (6,4029) FGCUUP(I1)                                      M0018044
0374       4029 FORMAT(40HOUNIT NUMBER FOR PERIPHERAL OPERATION IS,5X,A3)      M0018045
0375            WRITE (6,4030) FGCUUI(I1)                                      M0018046
0376       4030 FORMAT(40HOUNIT NUMBER FOR INTERMEDIATE STORAGE IS,5X,A3)      M0018047
0377            IFIXIT=FBLKFC(I1)                                              M0018048
0378            WRITE (6,4031) IFIXIT                                          M0018049
0379       4031 FORMAT(17HOBLOCKING FACTOR=,29X,I2)                            M0018050
0380            J1=FGCIPT(I1)                                                  M0018051
0381            IF(DEVCLS(J1)-D)5020,5032,5020                                 M0018052
0382       5032 IFIXIT=FLTNCY(I1)                                              M0018053
0383            WRITE (6,4032) IFIXIT                                          M0018054
0384       4032 FORMAT(22HOAVERAGE LATENCY TIME=,23X,I3)                       M0018055
0385            IFIXIT=FSEEK(I1)                                               M0018056
0386            WRITE (6,4033) IFIXIT                                          M0018057
0387       4033 FORMAT(19HOAVERAGE SEEK TIME=,26X,I3)                          M0018058
0388       5020 CONTINUE                                                       M0018059
          C
          C PRINT DEVICE INPUT SUMMARY
          C
0389            DO 5030 I1=1,MDEV                                              M0018060
0390            GO TO (5034,5033),I1                                           M0018061
0391       5033 IF(3*(I1/3)+1-I1)5001,5035,5001                                M0018062
0392       5034 WRITE (6,4034)                                                 M0018063
0393       4034 FORMAT(1H1,58X,16HDEVICE SUMMARIES//)                          M0018064
0394            GO TO 5001                                                     M0018065
0395       5035 WRITE (6,4035)                                                 M0018066
0396       4035 FORMAT(1H1,52X,28HDEVICE SUMMARIES (CONTINUED)//)              M0018067
```

```
      5001 IF(DEVCLS(I1)-D)5002,5003,5002                                M0018068
      5002 IF(DEVCLS(I1)-T)5005,5004,5005                                M0018069
      5003 IF(DEVMOD(I1)-BLANK4)5037,5036,5037                           M0018070
      5004 IF(DEVMOD(I1)-BLANK4)5039,5038,5039                           M0018071
      5005 IF(DEVMOD(I1)-BLANK4)5041,5040,5041                           M0018072
      5036 WRITE (6,4036) DEVTYP(I1)                                     M0018073
      4036 FORMAT(//5H TYPE,1X,A4,1X,4HDISK)                             M0018074
           GO TO 5006                                                    M0018075
      5037 WRITE (6,4037) DEVTYP(I1),DEVMOD(I1)                          M0018076
      4037 FORMAT(//5H TYPE,1X,A4,1X,5HMODEL,1X,A2,1X,4HDISK)            M0018077
           GO TO 5006                                                    M0018078
      5038 WRITE (6,4038) DEVTYP(I1)                                     M0018079
      4038 FORMAT(//5H TYPE,1X,A4,1X,4HTAPE)                             M0018080
           GO TO 5006                                                    M0018081
      5039 WRITE (6,4039) DEVTYP(I1),DEVMOD(I1)                          M0018082
      4039 FORMAT(//5H TYPE,1X,A4,1X,5HMODEL,1X,A2,1X,4HTAPE)            M0018083
           GO TO 5006                                                    M0018084
      5040 WRITE (6,4040) DEVTYP(I1)                                     M0018085
      4040 FORMAT(//5H TYPE,1X,A4,1X,18HUNIT RECORD DEVICE)              M0018086
           GO TO 5006                                                    M0018087
      5041 WRITE (6,4041) DEVTYP(I1),DEVMOD(I1)                          M0018088
      4041 FORMAT(//5H TYPE,1X,A4,1X,5HMODEL,1X,A2,1X,18HUNIT RECORD DEVICE) M0018089
      5006 IFIXIT=DEVSPD(I1)                                             M0018090
           IF(DEVCLS(I1)-U)5042,5043,5042                                M0018091
      5042 WRITE (6,4042) IFIXIT                                         M0018092
      4042 FORMAT(//7H SPEED=,38X,I7,1X,16HBYTES PER SECOND)             M0018093
           GO TO 5045                                                    M0018094
      5043 WRITE (6,4043) IFIXIT                                         M0018095
      4043 FORMAT(//7H SPEED=,41X,I4,1X,18HRECORDS PER MINUTE)           M0018096
           IFIXIT=DEVRCD(I1)                                             M0018097
           WRITE (6,4044) IFIXIT                                         M0018098
      4044 FORMAT(13H0RECORD SIZE=,38X,I3,1X,5HBYTES)                    M0018099
      5045 WRITE (6,4045) DEVSTP(I1)                                     M0018100
      4045 FORMAT(13H0SET UP TIME=,35X,F4.1,1X,7HMINUTES)                M0018101
           IF(DEVCLS(I1)-D)5007,5046,5007                                M0018102
      5007 IF(DEVCLS(I1)-T)5061,5048,5061                                M0018103
      5046 IFIXIT=DEVLSS(I1)                                             M0018104
           WRITE (6,4046) IFIXIT                                         M0018105
      4046 FORMAT(14H0LATENCY TIME=,35X,I3,1X,12HMILLISECONDS)           M0018106
           IFIXIT=DEVSRW(I1)                                             M0018107
           WRITE(6,4047) IFIXIT                                          M0018108
      4047 FORMAT(11H0SEEK TIME=,38X,I3,1X,12HMILLISECONDS)              M0018109
           GO TO 5061                                                    M0018110
      5048 FFIXIT=DEVLSS(I1)/10                                          M0018111
           WRITE (6,4048) FFIXIT                                         M0018112
      4048 FORMAT(17H0START-STOP TIME=,30X,F5.1,1X,12HMILLISECONDS)      M0018113
           FFIXIT=DEVSRW(I1)/10                                          M0018114
           WRITE (6,4049) FFIXIT                                         M0018115
```

```
0445          4049 FORMAT(13HOREWIND TIME=,34X,F5.1,1X,7HMINUTES)                MOO18115
0446               IFIXIT=DEVDEN(I1)                                             MOO18117
0447               WRITE (6,4060) IFIXIT                                         MOO18118
0448          4060 FORMAT(9HODENSITY=,40X,I3,1X,14HBYTES PER INCH)               MOO18119
0449          5061 WRITE (6,4061) DEVINT(I1)                                     MOO1811A
0450          4061 FORMAT(47HOAVERAGE MULTIPLEXOR CHANNEL INTERFERENCE TIME=,F5.1,1X,MOO18120
                  121HMICROSECONDS PER BYTE)                                     MOO18121
0451               K1=1                                                          MOO18122
0452               DO 5029 J1=1,MCUU                                             MOO18123
0453               IF(CUUPNT(J1)-I1)5029,5008,5029                               MOO18124
0454          5008 PRTCUU(K1)=CUU(J1)                                            MOO18125
0455               K1=K1+1                                                       MOO18126
0456               IF(K1-30)5029,5029,5062                                       MOO18127
0457          5062 K1=1                                                          MOO18128
0458               WRITE (6,4062) (PRTCUU(L1),L1=1,30)                           MOO18129
0459          4062 FORMAT(13HOUNIT NUMBERS,1X,30(A3,1X))                         MOO18130
0460          5029 CONTINUE                                                      MOO18131
0461               IF(K1-1)5030,5030,5063                                        MOO18132
0462          5063 K1=K1-1                                                       MOO18133
0463               WRITE (6,4062) (PRTCUU(L1),L1=1,K1)                           MOO18134
0464          5030 CONTINUE                                                      MOO18135
0465               I1=1                                                          MOO18200
0466               I2=1                                                          MOO18205
0467               I3=1                                                          MOO18210
0468               I4=3                                                          MOO18220
0469               I5=1                                                          MOO18230
0470               I6=1                                                          MOO18240
0471               I7=1                                                          MOO18245
0472               DAY=BLANK4                                                    MOO18253
0473               MMJOB=MWJOB                                                   MOO18260
              C
              C PROCESS JB, JF, JBD, JFD, JBF, JBIO, AND JFIO STATEMENTS
              C
0474           870 DO 179 IJ=I1,MMJOB                                            MOO18261
              C
              C DETERMINE WHICH STATEMENT IS BEING PROCESSED
              C
0475               GO TO (88,871,872,873,874,875,876,877,878,879,880),I7         MOO18263
0476           871 I7=1                                                          MOO18266
0477               GO TO 95                                                      MOO18269
0478           872 I7=1                                                          MOO18273
0479               GO TO 989                                                     MOO18276
0480           873 I7=1                                                          MOO18279
0481               GO TO 5068                                                    MOO18283
0482           874 I7=1                                                          MOO18286
0483               GO TO 154                                                     MOO18289
0484           875 I7=1                                                          MOO18293
0485               GO TO 192                                                     MOO18296
```

```
0486          875 I7=1                                                          MO018299
0487              GO TO 931                                                     MO018303
0488          877 I7=1                                                          MO018306
0489              GO TO 935                                                     MO018309
0490          878 I7=1                                                          MO018313
0491              GO TO 283                                                     MO018316
0492          879 I7=1                                                          MO018319
0493              GO TO 206                                                     MO018323
0494          880 I7=1                                                          MO018326
0495              GO TO 207                                                     MO018329
            C
            C READ A JB, JF, OR JZ STATEMENT
            C
0496           88 READ (5,89) A1,A2,A3,JOBNAM(IJ),STEPNM(IJ),DON(IJ),HON(IJ),    MO018400
                 1MON(IJ),DOFF(IJ),HOFF(IJ),MOFF(IJ),FIXINS,VARINS,PRECED,NOVLP  MO018500
0497           89 FORMAT(3A1,2X,2A8,I1,2I2,I1,2I2,2F5.0,A8,F2.2)                  MO018600
0498              IF(A1-J)1089,90,1089                                          MO018610
0499           90 IF(A2-Z)91,196,91                                             MO018620
            C
            C IS THIS STATEMENT FOR THE SAME DAY AS THE PREVIOUS STATEMENT
            C
0500           91 IF(ABS(A2)-ABS(DAY))1091,92,919                               MO018630
            C
            C YES.  CHECK BEGIN AND END TIMES FOR PROGRAM
            C
0501           92 CALL TOD (WEEK,DON(IJ),HON(IJ),MON(IJ),DOFF(IJ),HOFF(IJ),MOFF(IJ),MO018640
                 1HHON,MMON,HHOFF,MMOFF,ICASE,I3-1,ERROR,ERRORX)                  MO018650
            C
            C WERE THERE ANY ERRORS IN THE STATEMENT
            C
0502              IF(ERRORX-1)920,1092,1092                                     MO018655
            C
            C NO ERRORS IN STATEMENT.  IS THIS A DAILY OR WEEKLY PROGRAM
            C
0503          920 IF(ABS(A2)-ABS(D1))93,94,94                                   MO018660
            C
            C WEEKLY PROGRAM.  DO BEGIN AND END TIMES AGREE WITH PROGRAM TYPE
            C
0504           93 IF(ICASE-3)1093,95,1093                                       MO018670
            C
            C DAILY PROGRAM.  WAS TURNAROUND TIME OF PREVIOUS PROGRAM A FULL WORKING DAY
            C
0505           94 GO TO (940,941),I4                                            MO018680
            C
            C YES.  DO BEGIN AND END TIMES AGREE WITH PROGRAM TYPE
            C
0506          940 IF(ICASE-1)1940,95,1940                                       MO018690
            C
```

```
            C PARTIAL WORKING DAY. IS THE TURNAROUND TIME OF THIS PROGRAM A PARTIAL DAY
            C
0507          941 IF(TCASE-2)942,95,1093                                        MO018700
            C
            C NO. IS THE TURNAROUND TIME OF THIS PROGRAM A FULL WORKING DAY
            C
0508          942 IF(TCASE-1)1093,943,1093                                      MO018710
            C
            C WAS THE PREVIOUS PROGRAM A BACKGROUND PROGRAM OR A FOREGROUND PROGRAM
            C
0509           95 GO TO (97,96),I2                                              MO018720
            C
            C FOREGROUND.  IS THIS PROGRAM A FOREGROUND PROGRAM
            C
0510           96 IF(A3-F)1096,980,1096                                         MO018730
            C
            C PREVIOUS PROGRAM BACKGROUND.  IS THIS PROGRAM A BACKGROUND PROGRAM ALSO
            C
0511           97 IF(A3-B)981,980,981                                           MO018740
            C
            C NO.  IS THIS PROGRAM A FOREGROUND PROGRAM
            C
0512          981 IF(A3-F)1097,982,1097                                         MO018750
            C
            C YES.  SET SWITCH TO FOREGROUND
            C
0513          982 I2=2                                                          MO018760
0514              GO TO 980                                                     MO018770
            C
            C PREVIOUS PROGRAM TURNAROUND TIME WAS PARTIAL DAY, CURRENT PROGRAM FULL DAY
            C SAVE POINTER TO END OF PARTIAL WORKING DAY TABLE
            C MOVE PROGRAM DESCRIPTION TO FIRST FULL DAY PLACE IN TABLE
            C SET SWITCHES TO BACKGROUND, FULL WORKING DAY, AND PROGRAM TYPE CHANGE
            C
0515          943 MXDJOB=IJ-1                                                   MO018780
0516              MMMJOB=MJOB                                                   MO018790
0517              JJ=MDJOB+1                                                    MO018800
0518              CALL SWITCH (JOBNAM(IJ),STEPNM(IJ),DON(IJ),HON(IJ),MON(IJ),   MO018810
                 1DOFF(IJ),HOFF(IJ),MOFF(IJ),JOBNAM(JJ),STEPNM(JJ),DON(JJ),HON(JJ), MO018820
                 2MON(JJ),DOFF(JJ),HOFF(JJ),MOFF(JJ))                           MO018830
0519              IJ=JJ                                                         MO018840
0520              JJ=JJ+1                                                       MO018850
0521              I2=1                                                          MO018860
0522              I4=1                                                          MO018870
0523              I6=2                                                          MO018880
0524              GO TO 95                                                      MO018890
            C
            C THIS PROGRAM IS FOR A DIFFERRENT DAY THAN THE PREVIOUS PROGRAM
```

```
              C DETERMINE WHICH DAY IT IS AND SET NEXT DAY SWITCH TO THAT DAY
              C
  0525          910 GO TO (911,912,913,914,915,916,917,19101),I3                 M0018900
  0526          911 IF(ABS(A2)-ABS(D1))1911,918,921                              M0018910
  0527          912 IF(ABS(A2)-ABS(D2))1911,918,922                              M0018920
  0528          913 IF(ABS(A2)-ABS(D3))1911,918,923                              M0018930
  0529          914 IF(ABS(A2)-ABS(D4))1911,918,924                              M0018940
  0530          915 IF(ABS(A2)-ABS(D5))1911,918,925                              M0018950
  0531          916 IF(ABS(A2)-ABS(D6))1911,918,926                              M0018960
  0532          917 IF(ABS(A2)-ABS(D7))1911,918,1911                             M0018970
  0533          921 I5=2                                                         M0018980
  0534              GO TO 912                                                    M0018990
  0535          922 I5=3                                                         M0019000
  0536              GO TO 913                                                    M0019010
  0537          923 I5=4                                                         M0019020
  0538              GO TO 914                                                    M0019030
  0539          924 I5=5                                                         M0019040
  0540              GO TO 915                                                    M0019050
  0541          925 I5=6                                                         M0019060
  0542              GO TO 916                                                    M0019070
  0543          926 I5=7                                                         M0019080
  0544              GO TO 917                                                    M0019090
              C
              C IS IT A VALID DAY
              C
  0545          918 IF(I5-WEEK)919,919,1911                                      M0019100
              C
              C YES.  SAVE POINTERS TO THE BEGINNING AND END OF PARTIAL WORKING DAY TABLE
              C
  0546          919 MMMJOB=MOJOB                                                 M0019105
  0547              JJ=MWJOB+1                                                   M0019110
              C
              C WHAT WAS THE JOB TYPE OF THE PREVIOUS PROGRAM
              C
  0548              GO TO (927,928,929),I4                                       M0019120
              C
              C PREVIOUS PROGRAM WAS FULL WORKING DAY.  SAVE POINTER TO END OF THAT TABLE
              C
  0549          927 MXJOB=IJ-1                                                   M0019130
  0550              GO TO 930                                                    M0019140
              C
              C PREVIOUS PROGRAM WAS PARTIAL WORKING DAY.  SAVE POINTER TO THE END OF THAT
              C TABLE AND INDICATE THAT FULL WORKING DAY TABLE IS EMPTY
              C
  0551          928 MXDJOB=IJ-1                                                  M0019150
  0552              MXJOB=MOJOB                                                  M0019160
              C
              C SCHEDULE JOBS FOR THE PREVIOUS DAY
```

```
              C
0553             930 K1=I3-1                                                      MOO19200
0554                 I7=7                                                         MOO19250
0555                 GO TO 6000                                                   MOO19300
              C
              C PREVIOUS PROGRAM WAS WEEKLY.  SAVE POINTER TO THE END OF THAT TABLE
              C
0556             929 MXWJOB=IJ-1                                                  MOO19310
              C
              C MOVE PROGRAM DESCRIPTION TO FIRST PARTIAL DAY PLACE IN TABLE
              C
0557             931 CALL SWITCH (JOBNAM(IJ),STEPNM(IJ),DON(IJ),HON(IJ),MON(IJ),  MOO19320
                    1DOFF(IJ),HOFF(IJ),MOFF(IJ),JOBNAM(JJ),STEPNM(JJ),DON(JJ),HON(JJ), MOO19330
                    2MON(JJ),DOFF(JJ),HOFF(JJ),MOFF(JJ))                          MOO19340
              C
              C ANY DAYS SKIPPED BETWEEN CURRENT PROGRAM'S DAY AND PREVIOUS PROGRAM'S DAY
              C
0558                 IF(I5-I3-1)934,932,932                                       MOO19350
              C
              C YES.  INDICATE NULL TABLES AND SCHEDULE THOSE DAYS
              C
0559             932 MXDJOB=MWJOB                                                 MOO19360
0560                 MXJOB=MDJOB                                                  MOO19370
0561                 L1=I5-1                                                      MOO19380
0562                 K1=I3                                                        MOO19385
0563             933 I7=8                                                         MOO19390
0564                 GO TO 6000                                                   MOO19395
0565             935 K1=K1+1                                                      MOO19400
0566                 IF(K1-L1)933,933,934                                         MOO19405
              C
              C SET SWITHCES TO BACKGROUND, PARTIAL WORKING DAY, AND PROGRAM TYPE CHANGE
              C SET DAY EQUAL TO THIS PROGRAM'S DAY.  UPDATE NEXT DAY SWITCH
              C
0567             934 IJ=JJ                                                        MOO19500
0568                 JJ=JJ+1                                                      MOO19520
0569                 DAY=A2                                                       MOO19530
0570                 I5=I5+1                                                      MOO19540
0571                 I3=I5                                                        MOO19560
0572                 I2=1                                                         MOO19580
0573                 I4=2                                                         MOO19600
0574                 I6=2                                                         MOO19620
0575                 GO TO 92                                                     MOO19640
              C
              C CALCULATE OVERLAPPED AND NON-OVERLAPPED TIME ON CPU
              C
0576             980 TCPU(3,IJ,1)=(FIXINS*AVINST)*(1.-NOVLP)                      MOO19650
0577                 TNOVLP(3,IJ,1)=FIXINS*AVINST*NOVLP                           MOO19660
              C
```

```
      C IS THIS PROGRAM A FOREGROUND PROGRAM OR A BACKGROUND PROGRAM
      C
0578        GO TO (988,991),I2                                          MOO19670
      C
      C BACKGROUND PROGRAM.  IS THIS PROGRAM FOR THE SAME JOB AS THE PREVIOUS PROGRAM
      C
0579    988 GO TO (987,986,985),I4                                      MOO19671
0580    985 IL=IJ                                                       MOO19672
0581        GO TO 984                                                   MOO19673
0582    986 IL=IJ-MWJOB                                                 MOO19674
0583        GO TO 984                                                   MOO19675
0584    987 IL=IJ-MDJOB                                                 MOO19676
0585    984 GO TO (991,970),IL                                          MOO19677
0586    970 IK=IJ-1                                                     MOO19680
0587        IF(JOBNAM(IJ)-JOBNAM(IK))978,971,978                        MOO19690
      C
      C SAME JOB.  VERIFY STEP NAMES AND BEGIN AND END TIMES
      C
0588    971 IF(PRECED-STEPNM(IK))1971,972,1971                          MOO19700
0589    972 IF(DON(IJ)-DON(IK))1972,973,1972                            MOO19710
0590    973 IF(HON(IJ)-HON(IK))1972,974,1972                            MOO19720
0591    974 IF(MON(IJ)-MON(IK))1972,975,1972                            MOO19730
0592    975 IF(DOFF(IJ)-DOFF(IK))1972,976,1972                          MOO19740
0593    976 IF(HOFF(IJ)-HOFF(IK))1972,977,1972                          MOO19750
0594    977 IF(MOFF(IJ)-MOFF(IK))1972,991,1972                          MOO19760
      C
      C DIFFERENT JOBS.  CHECK FOR DUPLICATE JOB NAMES
      C
0595    978 PRECED=BLANK8                                               MOO19761
0596        IK=IK-1                                                     MOO19762
0597        DO 979 IX1=1,IK                                             MOO19764
0598        IF(JOBNAM(IJ)-JOBNAM(IX1))979,1978,979                      MOO19766
0599    979 CONTINUE                                                    MOO19768
      C
      C READ JBD OR JFD STATEMENTS
      C
0600    991 CALL FILE (D1,D2,J,A2,A3,MFILE,FILNAM,FILPNT,MCUU,CUU,CUUPNT, MOO19770
           1DEVCLS,BLANK4,BLANK8,D,ERROR,D3,D4)                         MOO19780
      C
      C IS THIS PROGRAM A BACKGROUND PROGRAM OR A FOREGROUND PROGRAM
      C
0601        GO TO (98,5068),I2                                          MOO19800
      C
      C BACKGROUND PROGRAM
      C
0602     98 I8=1                                                        MOO19850
0603    989 DO 114 IFG=I8,2                                             MOO19900
      C
```

```
             C READ A JBF STATEMENT
             C
0604              READ (5,99) A0,A4,A5,A6,A7,A8,FILEFG(IJ,IFG)                M0020000
0605            99 FORMAT(4A1,1X,2A1,A7)                                      M0020100
0606               IF(A0-J)1099,990,1099                                      M0020200
0607           990 IF(A4-A2)1990,101,1990                                     M0020250
0608           101 IF(A5-B)1101,102,1101                                      M0020300
0609           102 IF(A6-F)1102,103,1102                                      M0020400
             C
             C VERIFY FOREGROUND PROGRAM TYPE
             C
0610           103 IF(A7-BLANK4)105,104,105                                   M0020500
0611           104 IF(A8-BLANK4)1104,113,1104                                 M0020600
0612           105 IF(A7-P)107,106,107                                        M0020800
0613           106 IF(A8-BLANK4)1104,115,1104                                 M0020900
0614           107 IF(A7-R)109,108,109                                        M0021000
0615           108 IF(A8-BLANK4)1104,116,1104                                 M0021100
0616           109 IF(A7-T)1109,111,1109                                      M0021200
0617           111 IF(A8-I)112,117,112                                        M0021300
0618           112 IF(A8-O)1112,118,1112                                      M0021400
             C
             C THERE IS NO FOREGROUND PROGRAM REFERRED TO BY THIS STATEMENT
             C
0619           113 FLFGPT(IJ,IFG)=0                                           M0021500
0620               GO TO 114                                                  M0021600
             C
             C FOREGROUND PROGRAM IS A PRINTER TRANSCRIPTION PROGRAM
             C
0621           115 FLFGPT(IJ,IFG)=P                                           M0021700
0622               GO TO 119                                                  M0021800
             C
             C FOREGROUND PROGRAM IS A READER TRANSCRIPTION PROGRAM
             C
0623           116 FLFGPT(IJ,IFG)=R                                           M0021900
0624               GO TO 119                                                  M0022000
             C
             C FOREGROUND PROGRAM IS A TELECOMMUNICATIONS INPUT PROGRAM
             C
0625           117 FLFGPT(IJ,IFG)=I                                           M0022100
0626               GO TO 119                                                  M0022200
             C
             C FOREGROUND PROGRAM IS A TELECOMMUNICATIONS OUTPUT PROGRAM
             C
0627           118 FLFGPT(IJ,IFG)=0                                           M0022300
             C
             C WHICH FOREGROUND AREA HAS THIS FOREGROUND PROGRAM TYPE
             C
0628           119 IF(FLFGPT(IJ,IFG)-FG(1))121,120,121                        M0022400
```

```
0629        120 IF(FLEGPT(IJ,IFG)-FG(2))122,124,122                          M0022500
0630        121 IF(FLEGPT(IJ,IFG)-FG(2))1210,123,1210                        M0022600
          C
          C NO FOREGROUND AREA HAS THIS PROGRAM TYPE
          C
0631       1210 FLEGPT(IJ,IFG)=0                                              M0022650
0632            GO TO 114                                                     M0022670
          C
          C FOREGROUND AREA ONE
          C
0633        122 FLEGPT(IJ,IFG)=1                                              M0022700
0634            GO TO 125                                                     M0022800
          C
          C FOREGROUND AREA TWO
          C
0635        123 FLEGPT(IJ,IFG)=2                                              M0022900
0636            GO TO 125                                                     M0023000
          C
          C BOTH FOREGROUND AREAS
          C
0637        124 FLEGPT(IJ,IFG)=3                                              M0023100
          C
          C CHECK FOR VALID FILE NAME
          C
0638        125 IF(FILEFG(IJ,IFG)-BLANK8)1250,1125,1250                       M0023150
0639       1250 DO 126 J1=1,MFILE                                             M0023200
0640            IF(FILEFG(IJ,IFG)-FILNAM(J1))126,127,126                      M0023300
0641        126 CONTINUE                                                      M0023400
0642            GO TO 1126                                                    M0023450
0643        127 FLNMPT(IJ,IFG)=J1                                             M0023500
0644        114 CONTINUE                                                      M0023550
0645            IF(FLEGPT(IJ,1)*FLEGPT(IJ,2))1140,128,1140                    M0023570
          C
          C CHECK FOR UNIQUE FILE NAMES
          C
0646       1140 IF(FLNMPT(IJ,1)-FLNMPT(IJ,2))128,1127,128                     M0023600
          C
          C CAN EITHER OR BOTH FILES BE PROCESSED BY BOTH FOREGROUND PROGRAMS
          C
0647        128 IF(FLEGPT(IJ,1)-3)129,130,129                                 M0023620
0648        129 IF(FLEGPT(IJ,2)-3)1290,130,1290                               M0023640
          C
          C NO.  CHECK FOR UNIQUE FOREGROUND AREA ASSIGNMENTS
          C
0649       1290 IF(FLEGPT(IJ,1)-FLEGPT(IJ,2))5068,1291,5068                   M0023643
0650       1291 IF(FLEGPT(IJ,1)-3.)1129,5068,1129                             M0023646
          C
          C SAVE OVERLAPPED AND NON-OVERLAPPED CPU TIMES
```

```
             C
0651            130 TCPU(3,IJ,2)=TCPU(3,IJ,1)                                    M0023660
0652                TNOVLP(3,IJ,2)=TNOVLP(3,IJ,1)                                M0023680
             C
             C PRINT PROGRAM SUMMARY
             C
0653           5068 WRITE (6,4063)                                               M0023682
0654           4063 FORMAT(1H1,58X,15HPROGRAM SUMMARY////)                       M0023684
0655                GO TO (5064,5067),I2                                         M0023686
0656           5064 WRITE (6,4064) JOBNAM(IJ),STEPNM(IJ)                         M0023688
0657           4064 FORMAT(4H JOB,1X,A8,1X,4HSTEP,1X,A8)                         M0023690
0658                IF(PRECED-BLANK8)5065,5066,5065                              M0023692
0659           5065 WRITE (6,4065) PRECED                                        M0023694
0660           4065 FORMAT(17HOPRECEDED BY STEP,1X,A8)                           M0023696
0661           5066 WRITE (6,4066)                                               M0023698
0662           4066 FORMAT(27HOPROGRAM TYPE IS BACKGROUND)                       M0023700
0663                GO TO 5069                                                   M0023702
0664           5067 WRITE (6,4067) JOBNAM(IJ)                                    M0023704
0665           4067 FORMAT(8H PROGRAM,1X,A8)                                     M0023706
0666                WRITE (6,4068)                                               M0023708
0667           4068 FORMAT(27HOPROGRAM TYPE IS FOREGROUND)                       M0023710
0668           5069 GO TO (5072,5071,5070),I4                                    M0023712
0669           5070 WRITE (6,4070)                                               M0023714
0670           4070 FORMAT(27HOPROGRAM CATEGORY IS WEEKLY)                       M0023716
0671                GO TO 5073                                                   M0023718
0672           5071 WRITE (6,4071)                                               M0023720
0673           4071 FORMAT(40HOPROGRAM CATEGORY IS PARTIAL WORKING DAY)          M0023722
0674                GO TO 5073                                                   M0023724
0675           5072 WRITE (6,4072)                                               M0023726
0676           4072 FORMAT(37HOPROGRAM CATEGORY IS FULL WORKING DAY)             M0023728
0677           5073 WRITE (6,4073) DON(IJ),HON(IJ),MON(IJ)                       M0023730
0678           4073 FORMAT(29HOREADY FOR PROCESSING ON DAY,1X,I1,1X,2HAT,1X,I2,1X,9HHOM0023732
                   1URS AND,1X,I2,1X,7HMINUTES)                                  M0023734
0679                WRITE (6,4074) DOFF(IJ),HOFF(IJ),MOFF(IJ)                    M0023736
0680           4074 FORMAT(25HOMUST BE COMPLETED BY DAY,1X,I1,1X,2HAT,1X,I2,1X,9HHOURSM0023738
                   1 AND,1X,I2,1X,7HMINUTES)                                     M0023740
0681                IFIXIT=FIXINS                                                M0023742
0682                WRITE (6,4075) IFIXIT                                        M0023744
0683           4075 FORMAT(30HOFIXED NUMBER OF INSTRUCTIONS=,3X,I5)              M0023746
0684                IFIXIT=VARINS                                                M0023748
0685                WRITE (6,4076) IFIXIT                                        M0023750
0686           4076 FORMAT(33HOVARIABLE NUMBER OF INSTRUCTIONS=,I5,1X,10HPER RECORD)   M0023752
0687                IFIXIT=100*(NOVLP+.005)                                      M0023754
0688                WRITE (6,4077) IFIXIT                                        M0023756
0689           4077 FORMAT(29HONON-OVERLAPPED INSTRUCTIONS=,7X,I2,1X,30HPER CENT OF TOM0023758
                   1TAL INSTRUCTIONS)                                            M0023760
             C
             C PRINT FILE ASSIGNMENTS
```

```
                C
0690                  WRITE (6,4078)                                                   MO023763
0691            4078 FORMAT(////39H INPUT/OUTPUT UNIT ASSIGNMENTS//)                   MO023764
0692                  DO 5074 IFIXIT=1,MFILE                                           MO023766
0693                  IF(FILNAM(IFIXIT)-BLANK8)15079,15074,15079                       MO023767
0694           15079 IP1=FILPNT(IFIXIT)                                                MO023769
0695                  GO TO (5078,5079),I2                                             MO023770
0696            5079 WRITE (6,4079) FILNAM(IFIXIT),CUU(IP1)                            MO023772
0697            4079 FORMAT(5H FILE,1X,A7,1X,23HASSIGNED TO UNIT NUMBER,1X,A3)         MO023774
0698                  GO TO 5074                                                       MO023776
0699            5078 DO 5077 IK=1,2                                                    MO023778
0700                  IF(FG(IK)-BLANK4)5076,5077,5076                                  MO023780
0701            5076 IF(FILEFG(IJ,IK)-FILNAM(IFIXIT))5077,5074,5077                    MO023782
0702            5077 CONTINUE                                                          MO023784
0703                  GO TO 5079                                                       MO023786
0704            5074 CONTINUE                                                          MO023788
0705           15074 GO TO (15080,15075),I2                                            MO023789
0706           15080 DO 5075 IK=1,2                                                    MO023790
0707                  IF(FG(IK)-BLANK4)5080,5075,5080                                  MO023792
0708            5080 WRITE (6,4080) FILEFG(IJ,IK)                                      MO023794
0709            4080 FORMAT(5H0FILE,1X,A7,1X,24H*** SEE ASSIGNMENT BELOW)              MO023796
0710            5075 CONTINUE                                                          MO023798
0711           15075 IP1=0                                                             MO023799
                C
                C READ A JBIO OR JFIO STATEMENT
                C
0712             131 CALL FILEIO (A,D,H,I,O,T,U,D1,D2,BLANK8,J,A2,A3,MFILE,FILNAM,     MO023800
                    1FILPNT,CUUPNT,DEVCLS,DEVLSS,DEVSPW,VOLUME,RCDSIZ,BLKFCT,MULT,     MO023850
                    2LTNCY,SEEK,FACTOR,IOPNT,NAME,EIGHTY,ERROR,ERRORX,CLASS,DEVICE,D3, MO023900
                    3D4,D5)                                                            MO023950
                C
                C WERE THERE ANY ERRORS IN THE STATEMENT
                C
0713                  IF(ERRORX-1)132,1310,1310                                        MO024000
                C
                C ERRORS IN STATEMENT.  BYPASS TIMING SUBROUTINES
                C
0714            1310 ERRORX=0                                                          MO024030
0715                  GO TO 153                                                        MO024040
                C
                C NO ERRORS IN STATEMENT.  PRINT FILE DESCRIPTION SUMMARY
                C
0716             132 IP1=IP1+1                                                         MO024060
0717                  IF(3*(IP1/3)+1-IP1)5082,5081,5082                                MO024061
0718            5081 WRITE (6,4081)                                                    MO024062
0719            4081 FORMAT(1H1,52X,27HPROGRAM SUMMARY (CONTINUED))                    MO024063
0720            5082 WRITE (6,4082) NAME                                               MO024064
0721            4082 FORMAT(////5H FILE,1X,A7)                                         MO024065
```

```
0722              IF(MULT-X)5084,5083,5084                                     M0024066
0723         5083 WRITE (6,4083)                                               M0024067
0724         4083 FORMAT(70HONUMBER OF INSTRUCTIONS IN PROGRAM IS A FUNCTION OF THISM0024068
                 1 FILE'S VOLUME)                                              M0024069
0725         5084 IFIXIT=VOLUME                                                M0024070
0726              WRITE (6,4084) IFIXIT                                        M0024071
0727         4084 FORMAT(8HOVOLUME=,21X,I8)                                    M0024072
0728              IF(CLASS-U)5085,5088,5085                                    M0024073
0729         5085 IFIXIT=RCDSIZ                                                M0024074
0730              WRITE (6,4085) IFIXIT                                        M0024075
0731         4085 FORMAT(13HORECORD SIZE=,19X,I5,1X,5HBYTES)                   M0024076
0732              IFIXIT=BLKFCT                                                M0024077
0733              WRITE (6,4086) IFIXIT                                        M0024078
0734         4086 FORMAT(17HOBLOCKING FACTOR=,18X,I2)                          M0024079
0735              IF(CLASS-D)1320,5087,1320                                    M0024080
0736         5087 IFIXIT=LTNCY                                                 M0024081
0737              WRITE (6,4087) IFIXIT                                        M0024082
0738         4087 FORMAT(22HOAVERAGE LATENCY TIME=,12X,I3,1X,12HMILLISECONDS)  M0024083
0739              IFIXIT=SEEK                                                  M0024084
0740              WRITE (6,4088) IFIXIT                                        M0024085
0741         4088 FORMAT(19HOAVERAGE SEEK TIME=,15X,I3,1X,12HMILLISECONDS)     M0024086
0742              GO TO 1320                                                   M0024087
0743         5088 IFIXIT=DEVRCD(DEVICE)                                        M0024088
0744              WRITE (6,4085) IFIXIT                                        M0024089
0745              IFIXIT=1                                                     M0024089
0746              WRITE (6,4086) IFIXIT                                        M0024090
0747              IF(FACTOR-0.)1320,1320,5089                                  M0024091
0748         5089 WRITE (6,4089) FACTOR                                        M0024092
0749         4089 FORMAT(31HOFACTOR FOR UNIT RECORD DEVICE=,F6.3)              M0024093
            C
            C IS THIS A FOREGROUND PROGRAM OR A BACKGROUND PROGRAM
            C
0750         1320 GO TO (133,138),I2                                           M0024100
            C
            C BACKGROUND PROGRAM.  CHECK FOREGROUND FILES AGAINST CURRENT FILE NAME
            C
0751          133 DO 137 IFG=1,2                                               M0024200
0752              IF(FLFGPT(IJ,IFG)=0)137,137,134                              M0024250
0753          134 IF(NAME-FILEFG(IJ,IFG))137,135,137                           M0024300
            C
            C CURRENT FILE IS A FOREGROUND FILE.  IS ITS DEVICE CLASS UNIT RECORD
            C
0754          135 IF(CLASS-U)1135,136,1135                                     M0024400
            C
            C YES.  SAVE INFORMATION FROM JBIO STATEMENT AND BYPASS TIMING SUBROUTINES
            C
0755          136 FLFCTR(IFG)=FACTOR                                           M0024500
0756              FLIOPT(IFG)=IOPNT                                            M0024600
```

```
0757              FLDEV(IFG)=DEVICE                                              M0024700
0758              FLMULT(IFG)=MULT                                               M0024900
0759              FLVOL(IFG)=VOLUME                                              M0025000
0760              GO TO 153                                                      M0025100
0761          137 CONTINUE                                                       M0025200
        C
        C IS IT A DISK DEVICE
        C
0762          138 IF(CLASS-D)139,141,139                                         M0025300
        C
        C NOT A DISK DEVICE.  IS IT A TAPE OR A UNIT RECORD DEVICE
        C
0763          139 IF(CLASS-T)149,145,149                                         M0025500
        C
        C DISK DEVICE.  COMPUTE DEVICE, CHANNEL, CPU, AND SET UP TIMES
        C
0764          141 CALL DISKX (TDEV(3,IJ,IOPNT),TMPXBA(3,IJ,1),TMPXBW(3,IJ,1),    M0025600
                 1TSEL1A(3,IJ,1),TSEL1W(3,IJ,1),TSEL2A(3,IJ,1),TSEL2W(3,IJ,1),   M0025700
                 2TSEL3A(3,IJ,1),TSEL3W(3,IJ,1),TSEL4A(3,IJ,1),TSEL4W(3,IJ,1),   M0025800
                 3TSEL5A(3,IJ,1),TSEL5W(3,IJ,1),TSEL6A(3,IJ,1),TSEL6W(3,IJ,1),   M0025900
                 4TCPU(3,IJ,1),TSETUP(3,IJ,1),TNOVLP(3,IJ,1),DEVSPD(DEVICE),     M0026000
                 5DEVSTP(DEVICE),VOLUME,RCDSIZ,BLKFCT,MULT,LTNCY,SEEK,VARINS,AVINST,M0026100
                 6AVSEL,DEVINT(DEVICE),NOVLP,CUU(IOPNT),D1,D2,D3,D4,D5,D6,X)     M0026200
        C
        C IS THIS A BACKGROUND PROGRAM OR A FOREGROUND PROGRAM
        C
0765              GO TO (142,153),I2                                             M0026300
        C
        C BACKGROUND.  CAN BOTH FOREGROUND PROGRAMS BE USED FOR THE SAME FILE
        C
0766          142 IF(FLFGPT(IJ,1)-3)143,144,143                                  M0026400
0767          143 IF(FLFGPT(IJ,2)-3)153,144,153                                  M0026500
        C
        C YES.  COMPUTE DEVICE, CHANNEL, CPU, AND SET UP TIMES
        C
0768          144 CALL DISKX (SDEV(3,IJ,IOPNT),TMPXBA(3,IJ,2),TMPXBW(3,IJ,2),    M0026600
                 1TSEL1A(3,IJ,2),TSEL1W(3,IJ,2),TSEL2A(3,IJ,2),TSEL2W(3,IJ,2),   M0026700
                 2TSEL3A(3,IJ,2),TSEL3W(3,IJ,2),TSEL4A(3,IJ,2),TSEL4W(3,IJ,2),   M0026800
                 3TSEL5A(3,IJ,2),TSEL5W(3,IJ,2),TSEL6A(3,IJ,2),TSEL6W(3,IJ,2),   M0026900
                 4TCPU(3,IJ,2),TSETUP(3,IJ,2),TNOVLP(3,IJ,2),DEVSPD(DEVICE),     M0027000
                 5DEVSTP(DEVICE),VOLUME,RCDSIZ,BLKFCT,MULT,LTNCY,SEEK,VARINS,AVINST,M0027100
                 6AVSEL,DEVINT(DEVICE),NOVLP,CUU(IOPNT),D1,D2,D3,D4,D5,D6,X)     M0027200
0769              GO TO 153                                                      M0027300
        C
        C TAPE DEVICE.  COMPUTE DEVICE, CHANNEL, CPU, SET UP, AND REWIND TIMES
        C
0770          145 CALL TAPEX (TDEV(3,IJ,IOPNT),TMPXBA(3,IJ,1),TSEL1A(3,IJ,1),    M0027400
                 1TSEL2A(3,IJ,1),TSEL3A(3,IJ,1),TSEL4A(3,IJ,1),TSEL5A(3,IJ,1),   M0027500
```

```
               2TSEL6A(3,IJ,1),TCPU(3,IJ,1),TSETUP(3,IJ,1),TWNDUP(3,IJ,1),          M0027600
               3TNOVLP(3,IJ,1),DEVSPD(DEVICE),DEVSTP(DEVICE),VOLUME,RCDSIZ,BLKFCT,M0027700
               4MULT,DEVLSS(DEVICE),DEVSRW(DEVICE),DEVDEN(DEVICE),VARINS,AVINST,   M0027800
               5AVSEL,DEVINT(DEVICE),NOVLP,CUU(IOPNT),D1,D2,D3,D4,D5,D6,X)          M0027900
          C
          C IS THIS A BACKGROUND PROGRAM OR A FOREGROUND PROGRAM
          C
0771            GO TO (146,153),I2                                                  M0028000
          C
          C BACKGROUND.  CAN BOTH FOREGROUND PROGRAMS BE USED FOR THE SAME FILE
          C
0772        146 IF(FLFGPT(IJ,1)-3)147,148,147                                       M0028100
0773        147 IF(FLFGPT(IJ,2)-3)153,148,153                                       M0028200
          C
          C YES.  COMPUTE DEVICE, CHANNEL, CPU, SET UP, AND REWIND TIMES
          C
0774        148 CALL TAPEX (SDEV(3,IJ,IOPNT),TMPXBA(3,IJ,2),TSEL1A(3,IJ,2),         M0028300
               1TSEL2A(3,IJ,2),TSEL3A(3,IJ,2),TSEL4A(3,IJ,2),TSEL5A(3,IJ,2),        M0028400
               2TSEL6A(3,IJ,2),TCPU(3,IJ,2),TSETUP(3,IJ,2),TWNDUP(3,IJ,2),          M0028500
               3TNOVLP(3,IJ,2),DEVSPD(DEVICE),DEVSTP(DEVICE),VOLUME,RCDSIZ,BLKFCT,M0028600
               4MULT,DEVLSS(DEVICE),DEVSRW(DEVICE),DEVDEN(DEVICE),VARINS,AVINST,   M0028700
               5AVSEL,DEVINT(DEVICE),NOVLP,CUU(IOPNT),D1,D2,D3,D4,D5,D6,X)          M0028800
0775             GO TO 153                                                          M0028900
          C
          C UNIT RECORD DEVICE.  COMPUTE DEVICE, CHANNEL, CPU, AND SET UP TIMES
          C
0776        149 CALL UNITX (TDEV(3,IJ,IOPNT),TMPXMA(3,IJ,1),TSEL1A(3,IJ,1),         M0029000
               1TSEL2A(3,IJ,1),TSEL3A(3,IJ,1),TSEL4A(3,IJ,1),TSEL5A(3,IJ,1),        M0029100
               2TSEL6A(3,IJ,1),TCPU(3,IJ,1),TSETUP(3,IJ,1),TNOVLP(3,IJ,1),          M0029200
               3DEVSPD(DEVICE),DEVRCD(DEVICE),DEVSTP(DEVICE),VOLUME,MULT,FACTOR,   M0029300
               4VARINS,AVINST,AVSEL,DEVINT(DEVICE),NOVLP,CUU(IOPNT),D1,D2,D3,D4,   M0029400
               5D5,D6,X)                                                            M0029450
          C
          C IS THIS A BACKGROUND PROGRAM OR A FOREGROUND PROGRAM
          C
0777            GO TO (150,153),I2                                                  M0029500
          C
          C BACKGROUND.  CAN BOTH FOREGROUND PROGRAMS BE USED FOR THE SAME FILE
          C
0778        150 IF(FLFGPT(IJ,1)-3)151,152,151                                       M0029600
0779        151 IF(FLFGPT(IJ,2)-3)153,152,153                                       M0029700
          C
          C YES.  COMPUTE DEVICE, CHANNEL, CPU, AND SET UP TIMES
          C
0780        152 CALL UNITX (SDEV(3,IJ,IOPNT),TMPXMA(3,IJ,2),TSEL1A(3,IJ,2),         M0029800
               1TSEL2A(3,IJ,2),TSEL3A(3,IJ,2),TSEL4A(3,IJ,2),TSEL5A(3,IJ,2),        M0029900
               2TSEL6A(3,IJ,2),TCPU(3,IJ,2),TSETUP(3,IJ,2),TNOVLP(3,IJ,2),          M0030000
               3DEVSPD(DEVICE),DEVRCD(DEVICE),DEVSTP(DEVICE),VOLUME,MULT,FACTOR,   M0030100
```

```
                4VARINS,AVINST,AVSEL,DEVINT(DEVICE),NOVLP,FUH(IOPNT),D1,D2,D3,D4,  M0030200
               5D5,D6,X)                                                           M0030250
            C
            C CHECK FOR ADDITONAL JBTO OR JFTO STATEMENTS
            C
0781           153 IF(EIGHTY-BLANK4)131,1540,131                                   M0030300
0782          1540 I8=1                                                            M0030350
0783           154 IF(I8-MFILE)1541,1541,1550                                      M0030400
            C
            C CHECK THAT ALL FILES HAVE BEEN ASSIGNED TO A UNIT
            C
0784          1541 DO 155 J1=I8,MFILE                                              M0030450
0785                 IF(FILNAM(J1)-BLANK8)1154,155,1154                            M0030500
0786           155 CONTINUE                                                        M0030600
            C
            C IS THIS A BACKGROUND PROGRAM OR A FOREGROUND PROGRAM
            C
0787          1550 GO TO (156,1551),I2                                             M0030700
            C
            C FOREGROUND.  SET IDENTIFIER TO INDICATE THIS IS A FOREGROUND PROGRAM
            C
0788          1551 JCASE(IJ)=3                                                     M0030730
0789                 GO TO 270                                                     M0030760
            C
            C BACKGROUND.  CALCULATE DEVICE TIMES FOR FOREGROUND FILES
            C
0790           156 DO 277 IFG=1,2                                                  M0030800
0791                 FACTOR=FLFCTR(IFG)                                            M0030900
0792                 IOPNT=FLIOPT(IFG)                                             M0031000
0793                 DEVICE=FLDEV(IFG)                                             M0031100
0794                 MULT=FLMULT(IFG)                                              M0031200
0795                 VOLUME=FLVOL(IFG)                                             M0031300
            C
            C WHICH FOREGROUND PROGRAM CAN PROCESS THIS FILE
            C
0796                 IF(FLFGPT(IJ,IFG)-1)277,158,157                               M0031400
0797           157 IF(FLFGPT(IJ,IFG)-3)159,160,277                                 M0031500
            C
            C FOREGROUND PROGRAM ONE
            C
0798           158 K1=1                                                            M0031600
0799                 K2=1                                                          M0031700
0800                 K3=1                                                          M0031800
0801                 GO TO 161                                                     M0031900
            C
            C FOREGROUND PROGRAM TWO
            C
0802           159 K1=2                                                            M0032000
```

```
0803              K2=2                                                  M0032100
0804              K3=1                                                  M0032200
0805              GO TO 161                                             M0032300
          C
          C BOTH FOREGROUND PROGRAMS
          C
0806          160 K1=1                                                  M0032400
0807              K2=2                                                  M0032500
0808              K3=2                                                  M0032600
0809          161 DO 177 IFGX=K1,K2                                     M0032700
          C
          C IS THERE A DEVICE CONFLICT BETWEEN THE BACKGROUND AND THE FOREGROUND PROGRAMS
          C
0810              DO 164 IX=1,2                                         M0032800
0811              IF(FG(IX)-BLANK4)1610,164,1610                        M0032900
0812         1610 CUUP=FGUPPT(IX)                                       M0033000
0813              CIPT=FGCIPT(IX)                                       M0033050
0814              CUUI=FGUIPT(IX)                                       M0033100
0815              IF(TDEV(3,IJ,CUUP)-0.)176,162,176                     M0033200
0816          162 IF(DEVCLS(CIPT)-0)163,164,163                         M0033300
0817          163 IF(TDEV(3,IJ,CUUI)-0.)176,164,176                     M0033400
0818          164 CONTINUE                                              M0033410
          C
          C NO CONFLICT
          C
0819              JCASE(IJ)=1                                           M0033411
0820              GO TO (1640,1641),IJ                                  M0033412
0821         1641 IF(JOBNAM(IJ)-JOBNAM(IJ-1))1640,1642,1640             M0033413
0822         1642 IP1=JCASE(IJ-1)                                       M0033414
0823              GO TO (1640,1643,1640,1644),IP1                       M0033415
0824         1643 JCASE(IJ-1)=4                                         M0033416
0825         1644 JCASE(IJ)=4                                           M0033417
0826         1640 CIPT=FGCIPT(IFGX)                                     M0033420
0827              CPPT=FGCPPT(IFGX)                                     M0033440
0828              CUUI=FGUIPT(IFGX)                                     M0033460
0829              CUUP=FGUPPT(IFGX)                                     M0033480
          C
          C CAN BOTH FOREGROUND PROGRAMS PROCESS THIS FILE
          C
0830              GO TO (1650,169),K3                                   M0033500
          C
          C NO.  PRINT FILE ASSIGNMENT
          C
0831         1650 WRITE (6,4090) FILEFG(IJ,IFG),FGCUUT(IFGX)            M0033550
0832         4090 FORMAT(5H0FILE,1X,A7,1X,23HASSIGNED TO UNIT NUMBER,1X,A3)      M0033570
          C
          C IS THE INTERMEDIATE STORAGE DEVICE DISK OR TAPE
          C
```

```
0833         165 IF(DEVCLS(CIPT)-D)167,166,167                                  M0033600
           C
           C DISK DEVICE.  COMPUTE DEVICE, CHANNEL, CPU, AND SET UP TIMES
           C
0834         166 CALL DISKX (TDEV(IFGX,IJ,CUUI),TMPXBA(IFGX,IJ,1),             M0033700
                1TMPXBW(IFGX,IJ,1),TSEL1A(IFGX,IJ,1),TSEL1W(IFGX,IJ,1),         M0033800
                2TSEL2A(IFGX,IJ,1),TSEL2W(IFGX,IJ,1),TSEL3A(IFGX,IJ,1),         M0033900
                3TSEL3W(IFGX,IJ,1),TSEL4A(IFGX,IJ,1),TSEL4W(IFGX,IJ,1),         M0034000
                4TSEL5A(IFGX,IJ,1),TSEL5W(IFGX,IJ,1),TSEL6A(IFGX,IJ,1),         M0034100
                5TSEL6W(IFGX,IJ,1),TCPU(IFGX,IJ,1),TSETUP(IFGX,IJ,1),           M0034200
                6TNOVLP(IFGX,IJ,1),DEVSPD(CIPT),DEVSTP(CIPT),VOLUME,            M0034300
                7DEVRCD(CPPT)+1.,FBLKFC(IFGX),X,FLTNCY(IFGX),FSEEK(IFGX),       M0034400
                8FINSTX(IFGX),AVINST,AVSEL,DEVINT(CIPT),0.,CUU(CUUI),D1,D2,D3,D4,  M0034500
                9D5,D6,X)                                                       M0034550
0835              CALL DISKX (TDEV(3,IJ,CUUI),TMPXBA(3,IJ,1),TMPXBW(3,IJ,1),   M0034600
                1TSEL1A(3,IJ,1),TSEL1W(3,IJ,1),TSEL2A(3,IJ,1),TSEL2W(3,IJ,1),  M0034700
                2TSEL3A(3,IJ,1),TSEL3W(3,IJ,1),TSEL4A(3,IJ,1),TSEL4W(3,IJ,1),  M0034800
                3TSEL5A(3,IJ,1),TSEL5W(3,IJ,1),TSEL6A(3,IJ,1),TSEL6W(3,IJ,1),  M0034900
                4TCPU(3,IJ,1),TSETUP(3,IJ,1),TNOVLP(3,IJ,1),DEVSPD(CIPT),      M0035000
                5DEVSTP(CIPT),VOLUME,DEVRCD(CPPT)+1.,FBLKFC(IFGX),MULT,         M0035100
                6FLTNCY(IFGX),FSEEK(IFGX),VARINS,AVINST,AVSEL,DEVINT(CIPT),NOVLP,  M0035200
                7CUU(CUUI),D1,D2,D3,D4,D5,D6,X)                                M0035250
0836              GO TO 168                                                     M0035300
           C
           C TAPE DEVICE.  COMPUTE DEVICE, CHANNEL, CPU, SET UP, AND REWIND TIMES
           C
0837         167 CALL TAPEX (TDEV(IFGX,IJ,CUUI),TMPXBA(IFGX,IJ,1),             M0035400
                1TSEL1A(IFGX,IJ,1),TSEL2A(IFGX,IJ,1),TSEL3A(IFGX,IJ,1),         M0035500
                2TSEL4A(IFGX,IJ,1),TSEL5A(IFGX,IJ,1),TSEL6A(IFGX,IJ,1),         M0035600
                3TCPU(IFGX,IJ,1),TSETUP(IFGX,IJ,1),TWNDUP(IFGX,IJ,1),           M0035700
                4TNOVLP(IFGX,IJ,1),DEVSPD(CIPT),DEVSTP(CIPT),VOLUME,            M0035800
                5DEVRCD(CPPT)+1.,FBLKFC(IFGX),X,DEVLSS(CIPT),DEVSRW(CIPT),      M0035900
                6DEVDEN(CIPT),FINSTX(IFGX),AVINST,AVSEL,DEVINT(CIPT),0.,CUU(CUUI),  M0036000
                7D1,D2,D3,D4,D5,D6,X)                                           M0036050
0838              CALL TAPEX (TDEV(3,IJ,CUUI),TMPXBA(3,IJ,1),TSEL1A(3,IJ,1),   M0036100
                1TSEL2A(3,IJ,1),TSEL3A(3,IJ,1),TSEL4A(3,IJ,1),TSEL5A(3,IJ,1),  M0036200
                2TSEL6A(3,IJ,1),TCPU(3,IJ,1),TSETUP(3,IJ,1),TWNDUP(3,IJ,1),    M0036300
                3TNOVLP(3,IJ,1),DEVSPD(CIPT),DEVSTP(CIPT),VOLUME,DEVRCD(CPPT)+1.,  M0036400
                4FBLKFC(IFGX),MULT,DEVLSS(CIPT),DEVSRW(CIPT),DEVDEN(CIPT),VARINS,  M0036500
                5AVINST,AVSEL,DEVINT(CIPT),NOVLP,CUU(CUUI),D1,D2,D3,D4,D5,D6,X)  M0036600
           C
           C COMPUTE DEVICE, CHANNEL, CPU, AND SET UP TIMES FOR UNIT RECORD DEVICE
           C
0839         168 CALL UNITX (TDEV(IFGX,IJ,CHUP),TMPXMA(IFGX,IJ,1),             M0036700
                1TSEL1A(IFGX,IJ,1),TSEL2A(IFGX,IJ,1),TSEL3A(IFGX,IJ,1),         M0036800
                2TSEL4A(IFGX,IJ,1),TSEL5A(IFGX,IJ,1),TSEL6A(IFGX,IJ,1),         M0036900
                3TCPU(IFGX,IJ,1),TSETUP(IFGX,IJ,1),TNOVLP(IFGX,IJ,1),DEVSPD(CPPT),  M0037000
                4DEVRCD(CPPT),DEVSTP(CPPT),VOLUME,BLANK4,FACTOR,0.,AVINST,AVSEL,  M0037100
```

```
              5DEVINT(CPPT),0.,CUU(CUUP),D1,D2,D3,D4,D5,D6,X)             M0037200
0840          GO TO 177                                                   M0037300
        C
        C BOTH FOREGROUND PROGRAMS CAN PROCESS FILE.  SELECT PROPER TIMING SUBROUTINES
        C
0841      169 GO TO (170,171),IFG                                         M0037400
0842      170 GO TO (165,172),IFGX                                        M0037500
0843      171 GO TO (172,165),IFGX                                        M0037600
        C
        C IS THE INTERMEDIATE STORAGE DEVICE DISK OR TAPE.
        C
0844      172 IF(DEVCLS(CIPT)-0)174,173,174                               M0037700
        C
        C DISK DEVICE.  COMPUTE DEVICE, CHANNEL, CPU, AND SET UP TIMES
        C
0845      173 CALL DISKX (SDEV(IFGX,IJ,CUUT),TMPXBA(IFGX,IJ,2),           M0037800
             1TMPXBW(IFGX,IJ,2),TSEL1A(IFGX,IJ,2),TSEL1W(IFGX,IJ,2),      M0037900
             2TSEL2A(IFGX,IJ,2),TSEL2W(IFGX,IJ,2),TSEL3A(IFGX,IJ,2),      M0038000
             3TSEL3W(IFGX,IJ,2),TSEL4A(IFGX,IJ,2),TSEL4W(IFGX,IJ,2),      M0038100
             4TSEL5A(IFGX,IJ,2),TSEL5W(IFGX,IJ,2),TSEL6A(IFGX,IJ,2),      M0038200
             5TSEL6W(IFGX,IJ,2),TCPU(IFGX,IJ,2),TSETUP(IFGX,IJ,2),        M0038300
             6TNOVLP(IFGX,IJ,2),DEVSPD(CIPT),DEVSTP(CIPT),VOLUME,         M0038400
             7DEVRCD(CPPT)+1.,FBLKFC(IFGX),X,FLTNCY(IFGX),FSEEK(IFGX),    M0038500
             8FINSTX(IFGX),AVINST,AVSEL,DEVINT(CIPT),0.,CUU(CUUT),D1,D2,D3,D4, M0038600
             9D5,D6,X)                                                    M0038650
0846          CALL DISKX (SDEV(3,IJ,CUUT),TMPXBA(3,IJ,2),TMPXBW(3,IJ,2),  M0038700
             1TSEL1A(3,IJ,2),TSEL1W(3,IJ,2),TSEL2A(3,IJ,2),TSEL2W(3,IJ,2), M0038800
             2TSEL3A(3,IJ,2),TSEL3W(3,IJ,2),TSEL4A(3,IJ,2),TSEL4W(3,IJ,2), M0038900
             3TSEL5A(3,IJ,2),TSEL5W(3,IJ,2),TSEL6A(3,IJ,2),TSEL6W(3,IJ,2), M0039000
             4TCPU(3,IJ,2),TSETUP(3,IJ,2),TNOVLP(3,IJ,2),DEVSPD(CIPT),    M0039100
             5DEVSTP(CIPT),VOLUME,DEVRCD(CPPT)+1.,FBLKFC(IFGX),MULT,      M0039200
             6FLTNCY(IFGX),FSEEK(IFGX),VARINS,AVINST,AVSEL,DEVINT(CIPT),NOVLP, M0039300
             7CUU(CUUT),D1,D2,D3,D4,D5,D6,X)                              M0039350
0847          GO TO 175                                                   M0039400
        C
        C TAPE DEVICE.  COMPUTE DEVICE, CHANNEL, CPU, SET UP, AND REWIND TIMES
        C
0848      174 CALL TAPEX (SDEV(IFGX,IJ,CUUT),TMPXBA(IFGX,IJ,2),           M0039500
             1TSEL1A(IFGX,IJ,2),TSEL2A(IFGX,IJ,2),TSEL3A(IFGX,IJ,2),      M0039600
             2TSEL4A(IFGX,IJ,2),TSEL5A(IFGX,IJ,2),TSEL6A(IFGX,IJ,2),      M0039700
             3TCPU(IFGX,IJ,2),TSETUP(IFGX,IJ,2),TWNDUP(IFGX,IJ,2),        M0039800
             4TNOVLP(IFGX,IJ,2),DEVSPD(CIPT),DEVSTP(CIPT),VOLUME,         M0039900
             5DEVRCD(CPPT)+1.,FBLKFC(IFGX),X,DEVLSS(CIPT),DEVSRW(CIPT),   M0040000
             6DEVDEN(CIPT),FINSTX(IFGX),AVINST,AVSEL,DEVINT(CIPT),0.,CUU(CUUT), M0040100
             7D1,D2,D3,D4,D5,D6,X)                                        M0040150
0849          CALL TAPEX (SDEV(3,IJ,CUUT),TMPXBA(3,IJ,2),TSEL1A(3,IJ,2),  M0040200
             1TSEL2A(3,IJ,2),TSEL3A(3,IJ,2),TSEL4A(3,IJ,2),TSEL5A(3,IJ,2), M0040300
             2TSEL6A(3,IJ,2),TCPU(3,IJ,2),TSETUP(3,IJ,2),TWNDUP(3,IJ,2),  M0040400
```

```
                 3TNOVLP(3,IJ,2),DEVSPD(CIPT),DEVSTP(CIPT),VOLUME,DEVRCD(CPPT)+1.,  MO040500
                 4BLKFC(IFGX),MULT,DEVLSS(CIPT),DEVSPV(CIPT),DEVDEC(CIPT),VARINS,   MO040600
                 5AVINST,AVSEL,DEVINT(CIPT),NOVLP,CUU(CUUI),D1,D2,D3,D4,D5,D6,X)    MO040700
           C
           C COMPUTE DEVICE, CHANNEL, CPU, AND SET UP TIMES FOR UNIT RECORD DEVICE
           C
0850         175 CALL UNITX (SDEV(IFGX,IJ,CUUP),TMPXMA(IFGX,IJ,2),                    MO040800
                 1TSEL1A(IFGX,IJ,2),TSEL2A(IFGX,IJ,2),TSEL3A(IFGX,IJ,2),            MO040900
                 2TSEL4A(IFGX,IJ,2),TSEL5A(IFGX,IJ,2),TSEL6A(IFGX,IJ,2),            MO041000
                 3TCPU(IFGX,IJ,2),TSETUP(IFGX,IJ,2),TNOVLP(IFGX,IJ,2),DEVSPD(CPPT), MO041100
                 4DEVRCD(CPPT),DEVSTP(CPPT),VOLUME,BLANK4,FACTOR,0.,AVINST,AVSEL,   MO041200
                 5DEVINT(CPPT),0.,CUU(CUUP),D1,D2,D3,D4,D5,D6,X)                    MO041300
0851               GO TO 177                                                        MO041400
           C
           C DEVICE CONFLICT.  COMPUTE DEVICE, CHANNEL, CPU, AND SET UP TIMES FOR DEVICE
           C
0852         176 JCASE(IJ)=2                                                        MO041450
0853               GO TO (1740,1741),IJ                                             MO041451
0854        1741 IF(JOBNAM(IJ)-JOBNAM(IJ-1))1740,1742,1740                          MO041452
0855        1742 IP1=JCASE(IJ-1)                                                    MO041453
0856               GO TO (1743,1743,1740,1744),IP1                                  MO041454
0857        1743 JCASE(IJ-1)=4                                                      MO041455
0858        1744 JCASE(IJ)=4                                                        MO041456
0859        1740 CALL UNITX (TDEV(3,IJ,IOPNT),TMPXMA(3,IJ,1),TSEL1A(3,IJ,1),        MO041500
                 1TSEL2A(3,IJ,1),TSEL3A(3,IJ,1),TSEL4A(3,IJ,1),TSEL5A(3,IJ,1),      MO041600
                 2TSEL6A(3,IJ,1),TCPU(3,IJ,1),TSETUP(3,IJ,1),TNOVLP(3,IJ,1),        MO041700
                 3DEVSPD(DEVICE),DEVRCD(DEVICE),DEVSTP(DEVICE),VOLUME,MULT,FACTOR,  MO041800
                 4VARINS,AVINST,AVSEL,DEVINT(DEVICE),NOVLP,CUU(IOPNT),D1,D2,D3,D4,  MO041900
                 5D5,D6,X)                                                          MO041950
           C
           C PRINT FILE ASSIGNMENT
           C
0860               WRITE (6,4090) FILEFG(IJ,IFG),CUU(IOPNT)                         MO041955
0861         177 CONTINUE                                                           MO042000
0862         277 CONTINUE                                                           MO042010
           C
           C COMPUTE RESOURCE USE FRACTIONS FOR PROGRAM
           C
0863         270 CALL FRCTN (TOTAL(3,IJ,1),TCPU(3,IJ,1),TSEL1A(3,IJ,1),             MO042013
                 1TSEL1W(3,IJ,1),TSEL2A(3,IJ,1),TSEL2W(3,IJ,1),TSEL3A(3,IJ,1),      MO042016
                 2TSEL3W(3,IJ,1),TSEL4A(3,IJ,1),TSEL4W(3,IJ,1),TSEL5A(3,IJ,1),      MO042020
                 3TSEL5W(3,IJ,1),TSEL6A(3,IJ,1),TSEL6W(3,IJ,1),TMPXMA(3,IJ,1),      MO042023
                 4TMPXMW(3,IJ,1),TMPXBA(3,IJ,1),TMPXBW(3,IJ,1),TNOVLP(3,IJ,1),      MO042026
                 5CPU,BGFRCM,BGFRCB,BGFRCC)                                         MO042030
0864               DO 271 IL=1,12                                                   MO042033
0865               DO 271 IK=1,2                                                    MO042036
0866               FRACM(IJ,IL,IK)=BGFRCM(IL,IK)                                    MO042040
0867               FRACB(IJ,IL,IK)=BGFRCB(IL,IK)                                    MO042043
```

```
0868        271 FRACC(IJ,IL,IK)=BGFRCC(IL,IK)                                MO042046
0869            IL=JCASE(IJ)                                                 MO042048
          C
          C IS THIS A NON-CONFLICT BACKGROUND PROGRAM
          C
0870            GO TO (272,178,178),IL                                       MO042050
          C
          C YES. CAN BOTH FOREGROUND PROGRAMS PROCESS THE SAME FILE
          C
0871        272 GO TO (178,178,278,278,278,278,178,178,178,178),CASE        MO042053
          C
          C YES.  CALCULATE RESOURCE USE FRACTIONS
          C
0872        278 CALL FRCTN (TOTAL(3,IJ,2),TCPU(3,IJ,2),TSEL1A(3,IJ,2),      MO042056
               1TSEL1W(3,IJ,2),TSEL2A(3,IJ,2),TSEL2W(3,IJ,2),TSEL3A(3,IJ,2), MO042060
               2TSEL3W(3,IJ,2),TSEL4A(3,IJ,2),TSEL4W(3,IJ,2),TSEL5A(3,IJ,2), MO042063
               3TSEL5W(3,IJ,2),TSEL6A(3,IJ,2),TSEL6W(3,IJ,2),TMPXMA(3,IJ,2), MO042066
               4TMPXMW(3,IJ,2),TMPXBA(3,IJ,2),TMPXBW(3,IJ,2),TNOVLP(3,IJ,2), MO042070
               5CPU,BGFRCM,BGFRCB,BGFRCC)                                    MO042073
0873            DO 279 IL=1,12                                               MO042076
0874            DO 279 IK=1,2                                                MO042080
0875            FRACMX(IJ,IL,IK)=BGFRCM(IL,IK)                               MO042083
0876            FRACBX(IJ,IL,IK)=BGFRCB(IL,IK)                               MO042086
0877        279 FRACCX(IJ,IL,IK)=BGFRCC(IL,IK)                               MO042090
          C
          C DID THE PROGRAM TYPE CHANGE
          C
0878        178 GO TO (179,197),I6                                           MO042100
          C
          C NO.  READ STATEMENTS FOR NEXT PROGRAM
          C
0879        179 CONTINUE                                                     MO042200
          C
          C NO MORE ROOM IN TABLE FOR CURRENT PROGRAM TYPE.  DETERMINE PROGRAM TYPE
          C
0880            GO TO (183,182,181),I4                                       MO042300
          C
          C WEEKLY.  SAVE POINTER TO END OF TABLE AND SET DAY TO 1
          C
0881        181 MXWJOB=IJ                                                    MO042400
0882            DAY=D1                                                       MO042500
0883            GO TO 184                                                    MO042600
          C
          C PARTIAL WORKING DAY.  SAVE POINTER TO THE END OF THAT TABLE
          C SET DO LOOP LIMITS TO BEGINNING AND END OF FULL WORKING DAY TABLE
          C SET SWITCHES TO BACKGROUND AND FULL WORKING DAY
          C
0884        182 MXDJOB=IJ                                                    MO042700
```

```
0885          MMJOB=MJOB                                                      M0042750
0886          I1=MOJOB+1                                                      M0042800
0887          I2=1                                                            M0042850
0888          I4=1                                                            M0042900
0889          GO TO 870                                                       M0042950
        C
        C FULL WORKING DAY.  SAVE POINTER TO THE END OF THAT TABLE
        C
0890      183 MXJOB=IJ                                                        M0043000
        C
        C SCHEDULE JOBS FOR THE CURRENT DAY AND UPDATE CURRENT DAY SWITCH
        C
0891          I7=9                                                            M0043100
0892          K1=I3-1                                                         M0043110
0893          GO TO 6000                                                      M0043120
0894      283 IX3=I3-1                                                        M0043150
0895          GO TO (185,186,187,188,189,191,192),IX3                         M0043200
0896      185 DAY=D2                                                          M0043300
0897          GO TO 196                                                       M0043400
0898      186 DAY=D3                                                          M0043500
0899          GO TO 196                                                       M0043600
0900      187 DAY=D4                                                          M0043700
0901          GO TO 196                                                       M0043800
0902      188 DAY=D5                                                          M0043900
0903          GO TO 196                                                       M0044000
0904      189 DAY=D6                                                          M0044100
0905          GO TO 196                                                       M0044200
0906      191 DAY=D7                                                          M0044300
0907          GO TO 196                                                       M0044400
        C
        C READ JZ STATEMENT
        C
0908      192 READ (5,193) A1,A2                                              M0044500
0909      193 FORMAT(2A1)                                                     M0044600
0910          IF(A1-J)1193,194,1193                                           M0044700
0911      194 IF(A2-Z)1194,195,1194                                           M0044800
        C
        C END OF PROGRAM
        C
0912      195 CALL EXIT                                                       M0044900
        C
        C ANY MORE DAYS IN WEEK
        C
0913      196 IF(I5-WEEK)184,184,192                                          M0045000
        C
        C YES.  SET DO LOOP LIMITS TO BEGINNING AND END OF PARTIAL WORKING DAY TABLE
        C SET SWITCHES TO BACKGROUND AND PARTIAL WORKING DAY.  UPDATE NEXT DAY SWITCHES
        C
```

```
0914          184 MMJOB=MDJOB                                               M0045050
0915              I1=MWJOB+1                                                M0045100
0916              I2=1                                                      M0045200
0917              I3=I3+1                                                   M0045300
0918              I4=2                                                      M0045400
0919              I5=I5+1                                                   M0045500
0920              GO TO 870                                                 M0045700
            C
            C PROGRAM TYPE CHANGED.  SET DO LOOP LIMITS TO BEGINNING AND END OF CURRENT
            C TABLE AND RESET PROGRAM CHANGE SWITCH
            C
0921          197 MMJOB=MMMJOB                                              M0045800
0922              I1=JJ                                                     M0045900
0923              I6=1                                                      M0046000
0924              GO TO 870                                                 M0046100
            C
            C JZ STATEMENT READ.  DETERMINE PROGRAM TYPE OF PREVIOUS PROGRAM
            C
0925          198 GO TO (201,200,199),I4                                    M0046200
            C
            C WEEKLY.  SAVE POINTER TO END OF THAT TABLE
            C
0926          199 MXWJOB=IJ-1                                               M0046300
0927              GO TO 203                                                 M0046400
            C
            C PARTIAL WORKING DAY.  SAVE POINTER TO THE END OF THAT TABLE AND INDICATE THAT
            C FULL WOKING DAY TABLE IS EMPTY.
            C
0928          200 MXDJOB=IJ-1                                               M0046500
0929              MXJOB=MDJOB                                               M0046600
0930              GO TO 202                                                 M0046700
            C
            C FULL WORKING DAY.  SAVE POINTER TO THE END OF THAT TABLE
            C
0931          201 MXJOB=IJ-1                                                M0046800
            C
            C SCHEDULE JOBS FOR THE PREVIOUS DAY
            C
0932          202 I7=10                                                     M0046900
0933              K1=I3-1                                                   M0046910
0934              GO TO 6000                                                M0046920
            C
            C ARE THERE ANY MORE DAYS IN THE WEEK
            C
0935          206 IF(I3-WEEK)203,203,205                                    M0047000
            C
            C YES.  INDICATE NULL TABLES AND SCHEDULE THOSE DAYS
            C
```

```
0936         203 MXDJOB=MWJOB                                                    M0047100
0937             MXJOB=MDJOB                                                     M0047200
0938             K1=13                                                           M0047400
0939         204 I7=11                                                           M0047410
0940             GO TO 6000                                                      M0047420
0941         207 K1=K1+1                                                         M0047430
0942             IF(K1-WEEK)204,204,205                                          M0047440
           C
           C END OF PROGRAM
           C
0943         205 CALL EXIT                                                       M0047500
           C **************************************************************************
           C
           C SCHEDULE THE JOBS FOR THE CURRENT DAY
           C
           C **************************************************************************
           C
           C WERE THERE ANY ERRORS IN THE INPUT
           C
0944        6000 IF(ERROR-1)6999,9999,9999                                       M0050000
           C
           C NO.  SCHEDULE THE JOBS
           C
0945        6999 ISIDJ=MWJOB+1                                                   M0050050
0946             ISIJ=MDJOB+1                                                    M0050100
0947             BEGIN=60000000.*(60.*HON(K1)+MON(K1))                           M0050120
0948             END=60000000.*(60.*HON(K1)+MON(K1))                             M0050140
0949             LS=0                                                            M0050160
           C
           C ARE THERE ANY FULL WORKING DAY JOBS FOR THIS DAY
           C
0950             IF(MXJOB-MDJOB+1)9999,6001,6001                                 M0050200
           C
           C YES.  FIND OPTIMAL SEQUENCE FOR THE FULL WORKING DAY JOBS
           C
0951        6001 LS=1                                                            M0050300
           C
           C WHICH SEQUENCING CASE IS THIS
           C
0952             GO TO (6007,6007,6007,6007,6007,6007,6007,6008,6008,6010),CASE  M0050500
           C
           C THREE-MACHINE SEQUENCING PROBLEM
           C
0953        6007 DO 6070 IS=ISIJ,MXJOB                                           M0050600
           C
           C IS THIS A NON-CONFLICT BACKGROUND PROGRAM
           C
0954             IF(JCASE(IS)-2)6002,6070,6070                                   M0050700
```

```
              C
              C YES.  CALCULATE MULTIPROGRAMMING TIME
              C
 0955          6002 DO 6020 JS=1,12                                             M0050900
 0956               DO 6020 KS=1,2                                              M0051000
 0957               BGFRCM(JS,KS)=FRACM(IS,JS,KS)                               M0051100
 0958               BGFRCB(JS,KS)=FRACB(IS,JS,KS)                               M0051200
 0959          6020 BGFRCC(JS,KS)=FRACC(IS,JS,KS)                               M0051300
 0960               CALL TIME23 (FRCMF1,FRCBF1,FRCCF1,FRCMF2,FRCMB2,FRCCF2,PGFRCM,   M0051400
                   1BGFRCB,BGFRCC,FRCMMF,FRCMBF,FRCMCF,FRCBMF,FRCBBF,FRCBCF,FRCCMF,  M0051500
                   2FRCCBF,FRCCCF,2,CPU,TIMEF,TIMEC)                            M0051600
              C
              C ARE THERE PARALLEL OPERATIONS ON EITHER INPUT OR OUTPUT
              C
 0961               GO TO (6004,6004,6003,6003,6003,6003,6004),CASE             M0051700
              C
              C YES.  CALCULATE ALTERNATE MULTIPROGRAMMING TIME
              C
 0962          6003 DO 6030 JS=1,12                                             M0051800
 0963               DO 6030 KS=1,2                                              M0051900
 0964               BXFRCM(JS,KS)=FRACMX(IS,JS,KS)                              M0052000
 0965               BXFRCB(JS,KS)=FRACBX(IS,JS,KS)                              M0052100
 0966          6030 BXFRCC(JS,KS)=FRACCX(IS,JS,KS)                              M0052200
 0967               CALL TIME23 (FRCMF1,FRCBF1,FRCCF1,FRCMF2,FRCBF2,FRCCF2,BXFRCM,   M0052300
                   1BXFRCB,BXFRCC,FRCMMF,FRCMBF,FRCMCF,FRCBMF,FRCBBF,FRCBCF,FRCCMF,  M0052400
                   2FRCCBF,FRCCCF,2,CPU,TIMEF,TIMEX)                            M0052500
              C
              C SAVE LARGER TIME FOR USE IN SEQUENCING
              C
 0968               IF((TOTAL(3,IS,1)/TIMEC)-(TOTAL(3,IS,2)/TIMEX))6031,6004,6004    M0052600
 0969          6031 DO 6039 JS=1,12                                             M0052700
 0970               DO 6039 KS=1,2                                              M0052800
 0971               BGFRCM(JS,KS)=BXFRCM(JS,KS)                                 M0052900
 0972               BGFRCB(JS,KS)=BXFRCB(JS,KS)                                 M0053000
 0973          6039 BGFRCC(JS,KS)=BXFRCC(JS,KS)                                 M0053100
 0974               TIMEC=TIMEX                                                 M0053200
 0975               SCHED(3,IS,1)=TIMEC                                         M0053250
 0976               IF(IS-ISTJ-1)6038,6037,6037                                 M0053300
              C
              C IS THIS PROGRAM FOR THE SAME JOB AS THE PREVIOUS PROGRAM
              C
 0977          6037 IF(JOBNAM(IS)-JOBNAM(IS-1))6038,6139,6038                   M0053400
              C
              C NO.  WHICH PARALLEL OPERATION CASE IS THIS
              C
 0978          6038 GO TO (6033,6033,6033,6034,6035,6036),CASE                  M0053500
              C
              C INPUT.  SAME OPERATION.  SAME MACHINES
```

```
              C
0979           6033 QC(LS)=TOTAL(3,IS,2)/TIMEX+TSETUP(3,IS,2)+TWNDUP(3,IS,2)        MO053600
0980                GO TO 6144                                                     MO053700
              C
              C OUTPUT.  SAME OPERATION.  SAME MACHINES
              C
0981           6034 QA(LS)=TOTAL(3,IS,2)/TIMEX+TSETUP(3,IS,2)+TWNDUP(3,IS,2)        MO053800
0982                GO TO 6144                                                     MO053900
              C
              C INPUT.  SAME OPERATION.  DIFFERENT MACHINES
              C
0983           6035 QC(LS)=TOTAL(3,IS,2)/TIMEX+TSETUP(3,IS,2)+TWNDUP(3,IS,2)        MO054000
0984                GO TO 6146                                                     MO054100
              C
              C OUTPUT.  SAME OPERATION.  DIFFERENT MACHINES
              C
0985           6036 QA(LS)=TOTAL(3,IS,2)/TIMEX+TSETUP(3,IS,2)+TWNDUP(3,IS,2)        MO054200
0986                GO TO 6146                                                     MO054300
              C
              C THIS PROGRAM IS FOR THE SAME JOB AS THE PREVIOUS PROGRAM
              C
0987           6138 NS=LS-1                                                        MO054400
              C
              C WHICH PARALLEL OPERATION CASE IS THIS
              C
0988                GO TO (6133,6133,6133,6134,6135,6136),CASE                     MO054500
              C
              C INPUT.  SAME OPERATION.  SAME MACHINES
              C
0989           6133 QC(NS)=QC(NS)+TOTAL(3,IS,2)/TIMEX+TSETUP(3,IS,2)+TWNDUP(3,IS,2)  MO054600
0990                GO TO 6154                                                     MO054700
              C
              C OUTPUT.  SAME OPERATION.  SAME MACHINES
              C
0991           6134 QA(NS)=QA(NS)+TOTAL(3,IS,2)/TIMEX+TSETUP(3,IS,2)+TWNDUP(3,IS,2)  MO054800
0992                GO TO 6154                                                     MO054900
              C
              C INPUT.  SAME OPERATION.  DIFFERENT MACHINES
              C
0993           6135 QC(NS)=QC(NS)+TOTAL(3,IS,2)/TIMEX+TSETUP(3,IS,2)+TWNDUP(3,IS,2)  MO055000
0994                GO TO 6156                                                     MO055100
              C
              C OUTPUT,  SAME OPERATION DIFFERENT MACHINES
              C
0995           6136 QA(NS)=QA(NS)+TOTAL(3,IS,2)/TIMEX+TSETUP(3,IS,2)+TWNDUP(3,IS,2)  MO055200
0996                GO TO 6156                                                     MO055300
              C
              C NO PARALLEL OPERATIONS OR DIFFERENT OPERATIONS IN PARALLEL
```

```
                C
0997             6004 SCHED(3,IS,1)=TIMEC                                              M0055350
0998                  IF(IS-ISIJ-1)6040,6005,6005                                      M0055400
                C
                C IS THIS PROGRAM FOR THE SAME JOB AS THE PREVIOUS PROGRAM
                C
0999             6005 IF(JOBNAM(IS)-JOBNAM(IS-1))6040,6050,6040                        M0055500
                C
                C NO.  WHICH THREE MACHINE CASE IS THIS
                C
1000             6040 GO TO (6041,6042,6043,6044,6045,6046,6047),CASE                  M0055600
                C
                C PARALLEL OPERATIONS ON INPUT.  DIFFERENT OPERATIONS
                C
1001             6041 QA(LS)=TOTAL(1,IS,1)+TSETUP(1,IS,1)+TWNDUP(1,IS,1)               M0055700
1002                  QB(LS)=TOTAL(2,IS,1)/TIMEF+TSETUP(2,IS,1)+TWNDUP(2,IS,1)         M0055800
1003                  QC(LS)=TOTAL(3,IS,1)/TIMEC+TSETUP(3,IS,1)+TWNDUP(3,IS,1)         M0055900
1004                  GO TO 6069                                                       M0056000
                C
                C PARALLEL OPERATIONS ON OUTPUT.  DIFFERENT OPERATIONS
                C
1005             6042 QA(LS)=TOTAL(3,IS,1)/TIMEC+TSETUP(3,IS,1)+TWNDUP(3,IS,1)         M0056100
1006                  QB(LS)=TOTAL(1,IS,1)+TSETUP(1,IS,1)+TWNDUP(1,IS,1)               M0056200
1007                  QC(LS)=TOTAL(2,IS,1)/TIMEF+TSETUP(2,IS,1)+TWNDUP(2,IS,1)         M0056300
1008                  GO TO 6069                                                       M0056400
                C
                C PARALLEL OPERATIONS ON INPUT.  SAME OPERATION.  SAME MACHINES
                C
1009             6043 QC(NS)=QC(NS)+TOTAL(3,IS,1)/TIMEC+TSETUP(3,IS,1)+TWNDUP(3,IS,1)  M0056500
1010                  GO TO 6144                                                       M0056600
                C
                C PARALLEL OPERATIONS ON OUTPUT.  SAME OPERATION.  SAME MACHINES
                C
1011             6044 QA(LS)=TOTAL(3,IS,1)/TIMEC+TSETUP(3,IS,1)+TWNDUP(3,IS,1)         M0056700
1012             6144 QD(LS)=TOTAL(1,IS,1)+TSETUP(1,IS,1)+TWNDUP(1,IS,1)               M0056800
1013                  QDP(LS)=TOTAL(2,IS,1)/TIMEF+TSETUP(2,IS,1)+TWNDUP(2,IS,1)        M0056900
1014                  GO TO 6146 6069                                                  M0057000
                C
                C PARALLEL OPERATIONS ON INPUT.  SAME OPERATION.  DIFFERENT MACHINES
                C
1015             6045 QC(LS)=TOTAL(3,IS,1)/TIMEC+TSETUP(3,IS,1)+TWNDUP(3,IS,1)         M0057100
1016                  GO TO 6146                                                       M0057200
                C
                C PARALLEL OPERATIONS ON OUTPUT.  SAME OPERATION.  DIFFERENT MACHINES
                C
1017             6046 QA(LS)=TOTAL(3,IS,1)/TIMEC+TSETUP(3,IS,1)+TWNDUP(3,IS,1)         M0057300
1018             6146 QD(LS)=TOTAL(1,IS,1)+TSETUP(1,IS,1)+TWNDUP(1,IS,1)               M0057400
1019                  QDX(LS)=TOTAL(2,IS,2)/TIMEF+TSETUP(2,IS,2)+TWNDUP(2,IS,2)        M0057500
```

```
1020          ODP(LS)=TOTAL(1,IS,2)+TSETUP(1,IS,2)+TWNDUP(1,IS,2)               M0057600
1021          ODPX(LS)=TOTAL(2,IS,1)/TIMEF+TSETUP(2,IS,1)+TWNDUP(2,IS,1)         M0057700
1022          GO TO 6069                                                         M0057800
      C
      C NO PARALLEL OPERATIONS.  WHICH FOREGROUND AREA IS USED FOR INPUT
      C
1023     6047 IF(FG(1)-R)6147,6048,6147                                          M0057900
1024     6147 IF(FG(1)-I)6049,6048,6049                                          M0058000
      C
      C FOREGROUND AREA ONE USED FOR INPUT
      C
1025     6048 QA(LS)=TOTAL(1,IS,1)+TSETUP(1,IS,1)+TWNDUP(1,IS,1)                 M0058100
1026          QC(LS)=TOTAL(2,IS,1)/TIMEF+TSETUP(2,IS,1)+TWNDUP(2,IS,1)          M0058200
1027          GO TO 6149                                                         M0058300
      C
      C FOREGROUND AREA TWO USED FOR INPUT
      C
1028     6049 QA(LS)=TOTAL(2,IS,1)/TIMEF+TSETUP(2,IS,1)+TWNDUP(2,IS,1)          M0058400
1029          QC(LS)=TOTAL(1,IS,1)+TSETUP(1,IS,1)+TWNDUP(1,IS,1)                M0058500
1030     6149 QB(LS)=TOTAL(3,IS,1)/TIMEC+TSETUP(3,IS,1)+TWNDUP(3,IS,1)          M0058600
      C
      C THIS PROGRAM IS FOR THE SAME JOB AS THE PREVIOUS PROGRAM
      C
1031     6050 NS=LS-1                                                            M0058700
      C
      C WHICH THREE-MACHINE CASE IS THIS
      C
1032          GO TO (6051,6052,6053,6054,6055,6056,6057),CASE                    M0058800
      C
      C PARALLEL OPERATIONS ON INPUT.  DIFFERENT OPERATIONS
      C
1033     6051 QA(NS)=QA(NS)+TOTAL(1,IS,1)+TSETUP(1,IS,1)+TWNDUP(1,IS,1)         M0058900
1034          QB(NS)=QB(NS)+TOTAL(2,IS,1)/TIMEF+TSETUP(2,IS,1)+TWNDUP(2,IS,1)   M0059000
1035          QC(NS)=QC(NS)+TOTAL(3,IS,1)/TIMEC+TSETUP(3,IS,1)+TWNDUP(3,IS,1)   M0059100
1036          GO TO 6070                                                         M0059200
      C
      C PARALLEL OPERATIONS ON OUTPUT.  DIFFERENT OPERATIONS
      C
1037     6052 QA(NS)=QA(NS)+TOTAL(3,IS,1)/TIMEC+TSETUP(3,IS,1)+TWNDUP(3,IS,1)   M0059300
1038          QB(NS)=QB(NS)+TOTAL(1,IS,1)+TSETUP(1,IS,1)+TWNDUP(1,IS,1)         M0059400
1039          QC(NS)=QC(NS)+TOTAL(2,IS,1)/TIMEF+TSETUP(2,IS,1)+TWNDUP(2,IS,1)   M0059500
1040          GO TO 6070                                                         M0059600
      C
      C PARALLEL OPERATIONS ON INPUT.  SAME OPERATION.  SAME MACHINES
      C
1041     6053 QC(NS)=QC(NS)+TOTAL(3,IS,1)/TIMEC+TSETUP(3,IS,1)+TWNDUP(3,IS,1)   M0059700
1042          GO TO 6154                                                         M0059800
      C
```

```
              C PARALLEL OPERATIONS ON OUTPUT.  SAME OPERATION.  SAME MACHINES
              C
 1043           6054 QA(NS)=QA(NS)+TOTAL(3,IS,1)/TIMEC+TSETUP(3,IS,1)+TWNDUP(3,IS,1)     M0059900
 1044           6154 QD(NS)=QD(NS)+TOTAL(1,IS,1)+TSETUP(1,IS,1)+TWNDUP(1,IS,1)           M0060000
 1045                QDP(NS)=QDP(NS)+TOTAL(2,IS,1)/TIMEF+TSETUP(2,IS,1)+TWNDUP(2,IS,1)  M0060100
 1046                GO TO 6070                                                          M0060200
              C
              C PARALLEL OPERATIONS ON INPUT.  SAME OPERATION.  DIFFERENT MACHINES
              C
 1047           6055 QC(NS)=QC(NS)+TOTAL(3,IS,1)/TIMEC+TSETUP(3,IS,1)+TWNDUP(3,IS,1)     M0060300
 1048                GO TO 6156                                                          M0060400
              C
              C PARALLEL OPERATIONS ON OUTPUT.  SAME OPERATION.  DIFFERENT MACHINES
              C
 1049           6056 QA(NS)=QA(NS)+TOTAL(3,IS,1)/TIMEC+TSETUP(3,IS,1)+TWNDUP(3,IS,1)     M0060500
 1050           6156 QD(NS)=QD(NS)+TOTAL(1,IS,1)+TSETUP(1,IS,1)+TWNDUP(1,IS,1)           M0060600
 1051                QDP(NS)=QDP(NS)+TOTAL(1,IS,1)+TSETUP(1,IS,2)+TWNDUP(1,IS,2)         M0060700
 1052                QDX(NS)=QDX(NS)+TOTAL(2,IS,2)/TIMEF+TSETUP(2,IS,2)+TWNDUP(2,IS,2)  M0060800
 1053                QDPX(NS)=QDPX(NS)+TOTAL(2,IS,1)/TIMEF+TSETUP(2,IS,1)+               M0060900
                    1TWNDUP(2,IS,1)                                                      M0061000
 1054                GO TO 6070                                                          M0061100
              C
              C NO PARALLEL OPERATIONS.  WHICH FOREGROUND AREA IS USED FOR INPUT
              C
 1055           6057 IF(FG(1)-R)6157,6058,6157                                           M0061200
 1056           6157 IF(FG(1)-I)6059,6058,6059                                           M0061300
              C
              C FOREGROUND AREA ONE USED FOR INPUT
              C
 1057           6058 QA(NS)=QA(NS)+TOTAL(1,IS,1)+TSETUP(1,IS,1)+TWNDUP(1,IS,1)           M0061400
 1058                QC(NS)=QC(NS)+TOTAL(2,IS,1)/TIMEF+TSETUP(2,IS,1)+TWNDUP(2,IS,1)     M0061500
 1059                GO TO 6159                                                          M0061550
              C
              C FOREGROUND AREA TWO USED FOR INPUT
              C
 1060           6059 QA(NS)=QA(NS)+TOTAL(2,IS,1)/TIMEF+TSETUP(2,IS,1)+TWNDUP(2,IS,1)     M0061600
 1061                QC(NS)=QC(NS)+TOTAL(1,IS,1)+TSETUP(1,IS,1)+TWNDUP(1,IS,1)           M0061700
 1062           6159 QB(NS)=QB(NS)+TOTAL(3,IS,1)/TIMEC+TSETUP(3,IS,1)+TWNDUP(3,IS,1)     M0061800
 1063                GO TO 6070                                                          M0061900
 1064           6069 ISCHED(LS,2)=IS                                                     M0062000
 1065                LS=LS+1                                                             M0062100
              C
              C ARE THERE ANY MORE FULL WROKING DAY PROGRAMS
              C
 1066           6070 CONTINUE                                                            M0062300
 1067                LS=LS-1                                                             M0062500
              C
              C NO.  WERE THERE ANY NON-CONFLICT FULL WORKING DAY BACKGROUND PROGRAMS
```

```
               C
1068                 IF(LS-1)9999,6173,6073                                  MO062600
               C
               C YES.  FIND OPTIMAL SEQUENCE
               C
1069             6073 CALL IGNSCH (QA,QB,QC,QD,QDX,QDP,QDPX,QL,QN,LS,CASE,ERROR,ERRORX) MO062700
               C
               C COULD THE ALGORITHM FIND AN OPTIMAL SEQUENCE
               C
1070                 IF(ERRORX-1)6074,9999,9999                              MO062800
               C
               C YES.  SAVE SEQUENCING INFORMATION
               C
1071             6173 QL(1)=1                                                MO062900
1072                 QN(1)=1                                                 MO063000
1073             6074 DO 6076 IS=1,LS                                        MO063100
1074                 JS=QN(IS)                                               MO063200
1075                 ISCHED(IS,1)=ISCHED(JS,2)                               MO063300
1076                 KS=ISCHED(JS,2)                                         MO063400
1077                 NS=QL(IS)                                               MO063500
               C
               C WHICH THREE MACHINE CASE IS THIS
               C
1078                 GO TO (6081,6082,6083,6084,6085,6086,6087),CASE         MO063600
               C
               C PARALLEL OPERATIONS ON INPUT.  DIFFERENT OPERATIONS
               C
1079             6081 SCHED(1,KS,2)=QA(IS)                                   MO063700
1080                 SCHED(2,KS,2)=QB(IS)                                    MO063800
1081                 SCHED(3,KS,2)=QC(IS)                                    MO063900
1082                 GO TO 6078                                              MO064000
               C
               C PARALLEL OPERATIONS ON OUTPUT.  DIFFERENT OPERATIONS
               C
1083             6082 SCHED(3,KS,2)=QA(IS)                                   MO064100
1084                 SCHED(1,KS,2)=QB(IS)                                    MO064200
1085                 SCHED(2,KS,2)=QC(IS)                                    MO064300
1086                 GO TO 6078                                              MO064400
               C
               C PARALLEL OPERATIONS ON INPUT.  SAME OPERATIONS.  SAME MACHINES
               C
1087             6083 SCHED(3,KS,2)=QC(IS)                                   MO064500
1088                 GO TO (6094,6194),NS                                    MO064600
               C
               C PARALLEL OPERATIONS ON OUTPUT.  SAME OPERATIONS.  SAME MACHINES
               C
1089             6084 SCHED(3,KS,2)=QA(IS)                                   MO064700
1090                 GO TO (6094,6194),NS                                    MO064800
```

```
1091      6094 SCHED(1,KS,2)=QD(IS)                                  M0064900
1092           SCHED(2,KS,2)=QDP(IS)                                 M0065000
1093           GO TO 6079                                            M0065100
1094      6194 SCHED(1,KS,2)=QDP(IS)                                 M0065200
1095           SCHED(2,KS,2)=QD(IS)                                  M0065300
1096           GO TO 6079                                            M0065400
          C
          C PARALLEL OPERATIONS ON INPUT.  SAME OPERATION.  DIFFERENT MACHINES
          C
1097      6085 SCHED(3,KS,2)=QC(IS)                                  M0065500
1098           GO TO (6096,6196),NS                                  M0065600
          C
          C PARALLEL OPERATIONS ON OUTPUT.  SAME OPERATION.  DIFFERENT MACHINES
          C
1099      6086 SCHED(3,KS,2)=QA(IS)                                  M0065700
1100           GO TO (6096,6196),NS                                  M0065800
1101      6096 SCHED(1,KS,2)=QD(IS)                                  M0065900
1102           SCHED(2,KS,2)=QDPX(IS)                                M0066000
1103           GO TO 6079                                            M0066100
1104      6196 SCHED(1,KS,2)=QDP(IS)                                 M0066200
1105           SCHED(2,KS,2)=QDX(IS)                                 M0066300
1106           GO TO 6079                                            M0066400
          C
          C NO PARALLEL OPERATIONS.  WHCIH FOREGROUND AREA IS USED FOR INPUT
          C
1107      6087 IF(FG(1)-R)6187,6088,6187                             M0066500
1108      6187 IF(FG(1)-I)6089,6088,6089                             M0066600
          C
          C FOREGROUND AREA ONE USED FOR INPUT
          C
1109      6088 SCHED(1,KS,2)=QA(IS)                                  M0066700
1110           SCHED(2,KS,2)=QC(IS)                                  M0066800
1111           GO TO 6189                                            M0066900
          C
          C FOREGROUND AREA TWO USED FOR INPUT
          C
1112      6089 SCHED(1,KS,2)=QC(IS)                                  M0067000
1113           SCHED(2,KS,2)=QA(IS)                                  M0067100
1114      6189 SCHED(3,KS,2)=QB(IS)                                  M0067200
1115      6078 DO 6072 JS=1,3                                        M0067300
1116      6072 SCHED(JS,KS,3)=1                                      M0067400
1117           GO TO 6077                                            M0067500
1118      6079 SCHED(1,KS,3)=NS                                      M0067600
1119           SCHED(2,KS,3)=NS                                      M0067700
1120           SCHED(3,KS,3)=NS                                      M0067800
1121      6077 JCASE(KS)=0                                           M0067900
1122           DO 6075 JS=1,3                                        M0068000
1123           IF(SCHED(JS,KS,2))6075,6075,6175                      M0068100
```

```
1124        6175 JCASE(KS)=JCASE(KS)+JS                                          M0068200
1125        6075 CONTINUE                                                         M0068300
1126        6076 CONTINUE                                                         M0068400
1127             GO TO 6200                                                       M0068500
           C
           C TWO-MACHINE SEQUENCING PROBLEM.  WHICH FOREGROUND AREA IS BEING USED
           C
1128        6008 IF(FG(1)-BLANK4)6111,6112,6111                                   M0068600
           C
           C FOREGROUND AREA ONE IS BEING USED
           C
1129        6111 NS=1                                                             M0068700
1130             GO TO 6009                                                       M0068800
           C
           C FOREGROUND AREA TWO IS BEING USED
           C
1131        6112 NS=2                                                             M0068900
1132        6009 DO 6090 IS=ISTJ,MXJOB                                            M0069000
           C
           C IS THIS A NON-CONFLICT BACKGROUND PROGRAM
           C
1133             IF(JCASE(IS)-2)6113,6090,6090                                    M0069100
           C
           C YES.  CALCULATE MULTIPROGRAMMING TIME
           C
1134        6113 DO 6114 JS=1,12                                                  M0069200
1135             DO 6114 KS=1,2                                                   M0069300
1136             BGFRCM(JS,KS)=FRACM(IS,JS,KS)                                    M0069400
1137             BGFRCB(JS,KS)=FRACB(IS,JS,KS)                                    M0069500
1138        6114 BGFRCC(JS,KS)=FRACC(IS,JS,KS)                                    M0069600
1139             GO TO (6115,6116),NS                                             M0069700
1140        6115 CALL TIME23 (FRCMF1,FRCBF1,FRCCF1,BGFRCM,BGFRCB,BGFRCC,BGFRCM,   M0069800
                1BGFRCB,BGFRCC,FRCMMX,FRCMBX,FRCMCX,FRCBMX,FRCBBX,FRCBCX,FRCCMX,  M0069900
                2FRCCBX,FRCCCX,1,CPU,TIMEB,TIMEC)                                 M0070000
1141             GO TO 6117                                                       M0070100
1142        6116 CALL TIME23 (FRCMF2,FRCBF2,FRCCF2,BGFRCM,BGFRCB,BGFRCC,BGFRCM,   M0070200
                1BGFRCB,BGFRCC,FRCMMX,FRCMBX,FRCMCX,FRCBMX,FRCBBX,FRCBCX,FRCCMX,  M0070300
                2FRCCBX,FRCCCX,1,CPU,TIMEB,TIMEC)                                 M0070400
1143        6117 SCHED(3,IS,1)=TIMEB                                              M0070500
1144             IF(IS-ISIJ-1)6119,6118,6118                                      M0070600
           C
           C IS THIS PROGRAM FOR THE SAME JOB AS THE PREVIOUS PROGRAM
           C
1145        6118 IF(JOBNAM(IS)-JOBNAM(IS-1))6119,6123,6119                        M0070700
1146        6119 IP1=JCASE(IS)-7                                                  M0070800
           C
           C YES.  WHICH TWO MACHINE CASE IS THIS
           C
```

```
1147            GO TO (6121,6122),IP1                                         MOO70900
          C
          C FOREGROUND PROGRAM USED FOR INPUT
          C
1148       6121 QA(LS)=TOTAL(NS,IS,1)+TSETUP(NS,IS,1)+TWNDUP(NS,IS,1)          MOO71000
1149            QB(LS)=TOTAL(3,IS,1)/TIMEB+TSETUP(3,IS,1)+TWNDUP(3,IS,1)       MOO71100
1150            GO TO 6129                                                     MOO71200
          C
          C FOREGROUND PROGRAM USED FOR OUTPUT
          C
1151       6122 QA(LS)=TOTAL(3,IS,1)/TIMEB+TSETUP(3,IS,1)+TWNDUP(3,IS,1)       MOO71300
1152            QB(LS)=TOTAL(NS,IS,1)+TSETUP(NS,IS,1)+TWNDUP(NS,IS,1)          MOO71400
1153            GO TO 6129                                                     MOO71500
          C
          C THIS PROGRAM IS FOR THE SAME JOB AS THE PREVIOUS PROGRAM
          C
1154       6123 IP1=JCASE(IS)-7                                                MOO71600
1155            KS=LS-1                                                        MOO71700
          C
          C WHICH TWO MACHINE CASE IS THIS
          C
1156            GO TO (6124,6125),IP1                                          MOO71800
          C
          C FOREGROUND PROGRAM USED FOR INPUT
          C
1157       6124 QA(KS)=QA(KS)+TOTAL(NS,IS,1)+TSETUP(NS,IS,1)+TWNDUP(NS,IS,1)   MOO71900
1158            QB(KS)=QB(KS)+TOTAL(3,IS,1)/TIMEB+TSETUP(3,IS,1)+TWNDUP(3,IS,1) MOO72000
1159            GO TO 6090                                                     MOO72100
          C
          C FOREGROUND PROGRAM USED FOR OUTPUT
          C
1160       6125 QA(KS)=QA(KS)+TOTAL(3,IS,1)/TIMEB+TSETUP(3,IS,1)+TWNDUP(3,IS,1) MOO72200
1161            QB(KS)=QB(KS)+TOTAL(NS,IS,1)+TSETUP(NS,IS,1)+TWNDUP(NS,IS,1)   MOO72300
1162            GO TO 6090                                                     MOO72400
1163       6129 ISCHED(LS,2)=IS                                                MOO72500
1164            LS=LS+1                                                        MOO72600
          C
          C ARE THERE ANY MORE FULL WORKING DAY PROGRAMS
          C
1165       6090 CONTINUE                                                       MOO72700
1166            LS=LS-1                                                        MOO72800
          C
          C NO.  WERE THERE ANY NON-CONFLICT FULL WORKING DAY BACKGROUND PROGRAMS
          C
1167            IF(LS-1)9999,6127,6126                                         MOO72900
          C
          C YES.  FIND OPTIMAL SEQUENCE
          C
```

```
      6126 CALL JOHNSN(QA,QB,QN,LS)                                     M0073000
           GO TO 6128                                                   M0073100
C
C SAVE SEQUENCING INFORMATION
C
      6127 QN(1)=1                                                      M0073200
      6128 DO 6160 IS=1,LS                                              M0073300
           JS=QN(IS)                                                    M0073400
           ISCHED(IS,1)=ISCHED(JS,2)                                    M0073500
           KS=ISCHED(JS,2)                                              M0073600
           IP1=JCASE(IS)-7                                              M0073700
C
C WHICH TWO MACHINE CASE IS THIS
C
           GO TO (6161,6164),IP1                                        M0073800
      6161 GO TO (6162,6163),NS                                         M0073900
C
C FOREGROUND AREA ONE USED FOR INPUT
C
      6162 SCHED(1,KS,2)=QA(IS)                                         M0074000
           SCHED(2,KS,2)=0.                                             M0074100
           GO TO 6063                                                   M0074200
C
C FOREGROUND AREA TWO USED FOR INPUT
C
      6163 SCHED(1,KS,2)=0.                                             M0074300
           SCHED(2,KS,2)=QA(IS)                                         M0074400
      6063 SCHED(3,KS,2)=QB(IS)                                         M0074500
           GO TO 6067                                                   M0074600
      6164 GO TO (6165,6166),NS                                         M0074700
C
C FOREGROUND AREA ONE USED FOR OUTPUT
C
      6165 SCHED(1,KS,2)=QB(IS)                                         M0074800
           SCHED(2,KS,2)=0.                                             M0074900
           GO TO 6066                                                   M0075000
C
C FOREGROUND AREA TWO USED FOR OUTPUT
C
      6166 SCHED(1,KS,2)=0.                                             M0075100
           SCHED(2,KS,2)=QB(IS)                                         M0075200
      6066 SCHED(3,KS,2)=QA(IS)                                         M0075300
      6067 JCASE(KS)=0                                                  M0075400
           DO 6169 JS=1,3                                               M0075500
           SCHED(JS,KS,3)=1                                             M0075600
           IF(SCHED(JS,KS,2))6169,6169,6167                             M0075700
      6167 JCASE(KS)=JCASE(KS)+JS                                       M0075800
      6169 CONTINUE                                                     M0075900
```

```
1198      6160 CONTINUE                                                       MOC76000
1199           GO TO 6200                                                     MOC76100
         C
         C NO SEQUENCING PROBLEM
         C
1200      6010 DO 6110 IS=ISIJ,MXJOB                                          MOC76200
1201           IP1=JCASE(IS)                                                  MOC76300
1202           GO TO (6011,6011,6110,6011),IP1                                MOC76400
1203      6011 ISCHED(LS,1)=IS                                                MOC76500
1204           SCHED(3,IS,2)=TOTAL(3,IS,1)+TSETUP(3,IS,1)+TWNDUP(2,IS,1)      MOC76600
1205           SCHED(2,IS,2)=0.                                               MOC76700
1206           SCHED(1,IS,2)=0.                                               MOC76800
1207           DO 6019 KS=1,3                                                 MOC76900
1208      6019 SCHED(KS,IS,3)=1                                               MOC77000
1209           JCASE(IS)=3                                                    MOC77100
1210           LS=LS+1                                                        MOC77200
1211      6110 CONTINUE                                                       MOC77300
1212           LS=LS-1                                                        MOC77400
1213           GO TO 6210                                                     MOC77500
         C
         C ADJUST SEQUENCE FOR MULTISTEP JOBS
         C
1214      6200 NN=MXJOB                                                       MOC77550
1215           DO 6209 IS=ISIJ,MXJOB                                          MOC77600
         C
         C ARE THERE ANY MORE PROGRAMS SEQUENCED
         C
1216           IF(ISCHED(IS,1)-1)6210,6201,6201                               MOC77700
1217      6201 JS=ISCHED(IS,1)                                                MOC77800
         C
         C YES.  IS THIS PROGRAM FOR THE SAME JOB AS THE NEXT PROGRAM
         C
1218           IF(JOBNAM(JS)-JOBNAM(JS+1))6209,6202,6209                      MOC77900
1219      6202 ISIJ=IS+1                                                      MOC78000
         C
         C YES.  COMPUTE INDIVIDUAL PROGRAM MULTIPROGRAMMING TIMES
         C
1220           DO 6208 KS=JS,MXJOB                                            MOC78100
1221           MS=SCHED(3,KS,3)                                               MOC78200
1222           SCHED(3,KS,2)=TOTAL(3,KS,MS)/SCHED(2,KS,1)+TSETUP(3,KS,MS)+    MOC78300
              1TWNDUP(3,KS,MS)                                                MOC78400
1223           IF(JOBNAM(KS)-JOBNAM(KS+1))6207,6203,6207                      MOC78500
1224      6203 MS=IS+2                                                        MOC78600
1225           NS=IS+1                                                        MOC78700
1226           DO 6204 IP1=MS,MXJOB                                           MOC78800
1227      6204 ISCHED(MXJOB-IP1+MS,1)=ISCHED(MXJOB-IP1+NS,1)                  MOC78900
1228           ISCHED(IS+1,1)=JS+1                                            MOC79000
1229           ISIJ=IS+1                                                      MOC79100
```

```
          IS=IS+1                                                       MOC79150
     6208 LS=LS+1                                                       MOC79200
     6209 CONTINUE                                                      MOC79300
          GO TO 6210                                                    MOC79400
     6207 GO TO 6200                                                    MOC79500
C
C CALCULATE PROGRAM BEGIN AND END TIMES.  WHICH SEQUENCING CASE IS THIS
C
     6210 GO TO (6220,6240,6220,6240,6220,6240,6250,6220,6240,6230),CASE MOC79600
C
C FORFGROUND AREAS USED FOR INPUT
C
     6220 DO 6225 NS=1,2                                                MOC79700
          IS=1                                                          MOC79800
          JS=ISCHED(IS,1)                                               MOC79900
          SCHED(NS,JS,1)=BEGIN                                          MOC80000
          SCHED(NS,JS,2)=BEGIN+SCHED(NS,JS,2)                           MOC80050
          IF(SCHED(NS,JS,2)-END)6221,6221,6223                          MOC80100
     6221 SCHED(NS,JS,4)=1                                              MOC80150
          DO 6222 IS=2,LS                                               MOC80200
          JS=ISCHED(IS,1)                                               MOC80250
          KS=ISCHED(IS-1,1)                                             MOC80300
          SCHED(NS,JS,1)=SCHED(NS,KS,2)                                 MOC80350
          SCHED(NS,JS,2)=SCHED(NS,JS,1)+SCHED(NS,JS,2)                  MOC80400
          IF(SCHED(NS,JS,2)-END)6222,6222,6223                          MOC80450
     6222 SCHED(NS,JS,4)=1                                              MOC80500
          GO TO 6225                                                    MOC80550
     6223 SCHED(NS,JS,2)=SCHED(NS,JS,2)-SCHED(NS,JS,1)                  MOC80600
          SCHED(NS,JS,1)=0                                              MOC80650
          SCHED(NS,JS,4)=2                                              MOC80700
          MS=IS+1                                                       MOC80750
          DO 6224 IS=MS,LS                                              MOC80800
          JS=ISCHED(IS,1)                                               MOC80850
          SCHED(NS,JS,1)=0                                              MOC80900
     6224 SCHED(NS,JS,4)=2                                              MOC80950
     6225 CONTINUE                                                      MOC81050
          IS=1                                                          MOC81100
          JS=ISCHED(IS,1)                                               MOC81150
          SCHED(3,JS,1)=AMAX1(SCHED(1,JS,2),SCHED(2,JS,2))              MOC81200
          SCHED(3,JS,2)=SCHED(3,JS,1)+SCHED(3,JS,2)                     MOC81250
          IF(SCHED(3,JS,2)-END)6227,6227,6229                           MOC81300
     6227 DO 6228 IS=2,LS                                               MOC81350
          JS=ISCHED(IS,1)                                               MOC81400
          KS=ISCHED(IS-1,1)                                             MOC81450
          SCHED(3,JS,1)=AMAX1(SCHED(1,JS,2),SCHED(2,JS,2),SCHED(3,KS,2)) MOC81500
          SCHED(3,JS,2)=SCHED(3,JS,1)+SCHED(3,JS,2)                     MOC81550
          IF(SCHED(3,JS,2)-END)6228,6228,6229                           MOC81600
     6228 SCHED(3,JS,4)=1                                               MOC81650
```

```
1272           GO TO 9000                                                      MOC81700
1273      6229 SCHED(3,JS,2)=SCHED(3,JS,2)-SCHED(3,JS,1)                       MOC81750
1274           SCHED(3,JS,1)=0                                                 MOC81800
1275           SCHED(3,JS,4)=2                                                 MOC81850
1276           MS=IS+1                                                         MOC81900
1277           DO 6226 IS=MS,LS                                                MOC81950
1278           JS=ISCHED(IS,1)                                                 MOC82000
1279           SCHED(3,JS,1)=0                                                 MOC82050
1280      6226 SCHED(3,JS,4)=2                                                 MOC82100
1281           GO TO 9000                                                      MOC82150
        C
        C FOREGROUND AREAS USED FOR OUTPUT
        C
1282      6240 IS=1                                                            MOC82200
1283           JS=ISCHED(IS,1)                                                 MOC82250
1284           SCHED(3,JS,1)=BEGIN                                             MOC82300
1285           SCHED(3,JS,2)=BEGIN+SCHED(3,JS,2)                               MOC82350
1286           IF(SCHED(3,JS,2)-END)6241,6241,6243                             MOC82400
1287      6241 DO 6242 IS=2,LS                                                 MOC82450
1288           JS=ISCHED(IS,1)                                                 MOC82500
1289           KS=ISCHED(IS-1,1)                                               MOC82550
1290           SCHED(3,JS,1)=SCHED(3,KS,2)                                     MOC82600
1291           SCHED(3,JS,2)=SCHED(3,JS,1)+SCHED(3,JS,2)                       MOC82650
1292           IF(SCHED(3,JS,2)-END)6242,6242,6243                             MOC82700
1293      6242 SCHED(3,JS,4)=1                                                 MOC82750
1294           GO TO 6245                                                      MOC82800
1295      6243 SCHED(3,JS,2)=SCHED(3,JS,2)-SCHED(3,JS,1)                       MOC82850
1296           SCHED(3,JS,1)=0                                                 MOC82900
1297           SCHED(3,JS,4)=2                                                 MOC82950
1298           MS=IS+1                                                         MOC83000
1299           DO 6244 IS=MS,LS                                                MOC83050
1300           SCHED(3,JS,1)=0                                                 MOC83100
1301      6244 SCHED(3,JS,4)=2                                                 MOC83150
1302      6245 DO 6239 NS=1,2                                                  MOC83200
1303           IS=1                                                            MOC83250
1304           JS=ISCHED(IS,1)                                                 MOC83300
1305           SCHED(NS,JS,1)=SCHED(3,JS,2)                                    MOC83350
1306           SCHED(NS,JS,2)=SCHED(NS,JS,1)+SCHED(NS,JS,2)                    MOC83400
1307           IF(SCHED(NS,JS,2)-END)6246,6246,6248                            MOC83450
1308      6246 SCHED(NS,JS,4)=1                                                MOC83500
1309           DO 6247 IS=2,LS                                                 MOC83550
1310           JS=ISCHED(IS,1)                                                 MOC83600
1311           KS=ISCHED(IS-1,1)                                               MOC83650
1312           SCHED(NS,JS,1)=AMAX1(SCHED(3,JS,2),SCHED(NS,KS,2))              MOC83700
1313           SCHED(NS,JS,2)=SCHED(NS,JS,1)+SCHED(NS,JS,2)                    MOC83750
1314           IF(SCHED(NS,JS,2)-END)6247,6247,6248                            MOC83800
1315      6247 SCHED(NS,JS,4)=1                                                MOC83850
1316           GO TO 9000                                                      MOC83900
```

```
1317      6248 SCHED(NS,JS,2)=SCHED(NS,JS,2)-SCHED(NS,JS,1)                  MOC83950
1318           SCHED(NS,JS,1)=0                                              MOC84000
1319           SCHED(NS,JS,4)=2                                              MOC84050
1320           MS=IS+1                                                       MOC84100
1321           DO 6249 IS=MS,LS                                              MOC84150
1322           JS=ISCHED(IS,1)                                               MOC84200
1323           SCHED(NS,JS,1)=0                                              MOC84250
1324      6249 SCHED(NS,JS,4)=2                                              MOC84300
1325      6239 CONTINUE                                                      MOC84350
1326           GO TO 9000                                                    MOC84400
         C
         C FOREGROUND AREAS NOT USED
         C
1327      6230 DO 6235 NS=1,3                                                MOC84450
1328           IS=1                                                          MOC84500
1329           JS=ISCHED(IS,1)                                               MOC84550
1330           SCHED(NS,JS,1)=BEGIN                                          MOC84600
1331           SCHED(NS,JS,2)=BEGIN+SCHED(NS,JS,2)                           MOC84650
1332           IF(SCHED(NS,JS,2)-END)6231,6231,6233                          MOC84700
1333      6231 SCHED(NS,JS,4)=1                                              MOC84750
1334           DO 6232 IS=2,LS                                               MOC84800
1335           JS=ISCHED(IS,1)                                               MOC84850
1336           KS=ISCHED(IS-1,1)                                             MOC84900
1337           SCHED(NS,JS,1)=SCHED(NS,KS,2)                                 MOC84950
1338           SCHED(NS,JS,2)=SCHED(NS,JS,1)+SCHED(NS,JS,2)                  MOC85000
1339           IF(SCHED(NS,JS,2)-END)6232,6232,6233                          MOC85050
1340      6232 SCHED(NS,JS,4)=1                                              MOC85100
1341           GO TO 6235                                                    MOC85150
1342      6233 SCHED(NS,JS,2)=SCHED(NS,JS,2)-SCHED(NS,JS,1)                  MOC85200
1343           SCHED(NS,JS,1)=0                                              MOC85250
1344           SCHED(NS,JS,4)=2                                              MOC85300
1345           MS=IS+1                                                       MOC85350
1346           DO 6234 IS=MS,LS                                              MOC85400
1347           JS=ISCHED(IS,1)                                               MOC85450
1348           SCHED(NS,JS,1)=0                                              MOC85500
1349      6234 SCHED(NS,JS,4)=2                                              MOC85550
1350      6235 CONTINUE                                                      MOC85600
1351           GO TO 9000                                                    MOC85650
         C
         C FOREGROUND AREAS USED FOR BOTH INPUT AND OUTPUT.  WHICH AREA IS USED FOR INPUT
         C
1352      6250 IF(FG(1)-R)6236,6237,6236                                     MOC85700
1353      6236 IF(FG(1)-I)6238,6237,6238                                     MOC85750
         C
         C FOREGROUND AREA ONE USED FOR INPUT
         C
1354      6237 NS=1                                                          MOC85800
1355           GO TO 6255                                                    MOC85850
```

```
      C
      C FOREGROUND AREA TWO USED FOR INPUT
      C
       6238 NS=2                                                        MOC85900
       6255 IS=1                                                        MOC85950
            JS=ISCHED(IS,1)                                             MOC86000
            SCHED(NS,JS,1)=BEGIN                                        MOC86050
            SCHED(NS,JS,2)=BEGIN+SCHED(NS,JS,2)                         MOC86100
            IF(SCHED(NS,JS,2)-END)6251,6251,6253                        MOC86150
       6251 SCHED(NS,JS,4)=1                                            MOC86200
            DO 6252 IS=2,LS                                             MOC86250
            JS=ISCHED(IS,1)                                             MOC86300
            KS=ISCHED(IS-1,1)                                           MOC86350
            SCHED(NS,JS,1)=SCHED(NS,KS,2)                               MOC86400
            IF(SCHED(NS,JS,2)-END)6252,6252,6253                        MOC86450
       6252 SCHED(NS,JS,4)=1                                            MOC86500
            GO TO 6256                                                  MOC86550
       6253 SCHED(NS,JS,2)=SCHED(NS,JS,2)-SCHED(NS,JS,1)                MOC86600
            SCHED(NS,JS,1)=0                                            MOC86650
            SCHED(NS,JS,4)=2                                            MOC86700
            MS=IS+1                                                     MOC86750
            DO 6254 IS=MS,LS                                            MOC86800
            JS=ISCHED(IS,1)                                             MOC86850
            SCHED(NS,JS,1)=0                                            MOC86900
       6254 SCHED(NS,JS,4)=2                                            MOC86950
       6256 IP1=3-NS                                                    MOC87000
            DO 6260 NN=IP1,3,NS                                         MOC87050
            GO TO (6261,6261,6262),NN                                   MOC87100
       6261 IK=NS                                                       MOC87150
            GO TO 6263                                                  MOC87200
       6262 IK=3-NS                                                     MOC87250
       6263 IS=1                                                        MOC87300
            JS=ISCHED(IS,1)                                             MOC87350
            SCHED(NN,JS,1)=SCHED(IK,JS,2)                               MOC87400
            SCHED(NN,JS,2)=SCHED(NN,JS,1)+SCHED(NN,JS,2)                MOC87450
            IF(SCHED(NN,JS,2)-END)6257,6257,6259                        MOC87500
       6257 DO 6258 IS=2,LS                                             MOC87550
            JS=ISCHED(IS,1)                                             MOC87600
            KS=ISCHED(IS-1,1)                                           MOC87650
            SCHED(NN,JS,1)=AMAX1(SCHED(IK,JS,2),SCHED(NN,KS,2))         MOC87700
            IF(SCHED(NN,JS,2)-END)6258,6258,6259                        MOC87750
       6258 SCHED(NN,JS,4)=1                                            MOC87800
            GO TO 9000                                                  MOC87850
       6259 SCHED(NN,JS,2)=SCHED(NN,JS,2)-SCHED(NN,JS,1)                MOC87900
            SCHED(NN,JS,1)=0                                            MOC87950
            SCHED(NN,JS,4)=2                                            MOC88000
            MS=IS+1                                                     MOC88050
            DO 6264 IS=MS,LS                                            MOC88100
```

```
1401              JS=ISCHED(IS,1)                                              MOC88150
1402              SCHED(NN,JS,1)=0                                             MOC88200
1403         6264 SCHED(NN,JS,4)=2                                             MOC88250
1404         6260 CONTINUE                                                     MOC88300
           C
           C ARE THERE PARALLEL OPERATIONS ON EITHER INPUT OR OUTPUT
           C
1405         9000 GO TO (9900,9900,9300,9300,9300,9300,9900,9900,9900,9900),CASE   MOC88350
           C
           C YES.  PRINT FOREGROUND FILE ASSIGNMENTS
1406         9300 WRITE (6,9301)                                               MOC88400
1407         9301 FORMAT(1H1,52X,27HFOREGROUND FILE ASSIGNMENTS//)             MOC88450
1408              WRITE (6,9302) FGCUU(1),FGCUU(2)                             MOC88500
1409         9302 FORMAT(1H0,18HBACKGROUND PROGRAM,8X,2(5X,21HFILE ASSIGNED TO UNIT,MOC88550
                 11X,A3))                                                      MOC88555
1410              DO 9309 IS=1,LS                                              MOC88600
1411              JS=ISCHED(IS,1)                                              MOC88650
1412              MS=SCHED(3,JS,3)                                             MOC88700
1413              GO TO (9303,9304),MS                                         MOC88750
1414         9303 WRITE (6,9305) JOBNAM(JS),STEPNM(JS),FILEFG(JS,1),FILEFG(JS,2)   MOC88800
1415              GO TO 9309                                                   MOC88850
1416         9304 WRITE (6,9305) JOBNAM(JS),STEPNM(JS),FILEFG(JS,2),FILEFG(JS,1)   MOC88900
1417         9305 FORMAT(1H0,3HJOB,1X,A8,1X,4HSTEP,1X,A8,5X,A7,23X,A7)         MOC88950
1418         9309 CONTINUE                                                     MOC89000
           C
           C PRINT SCHEDULES FOR THE CURRENT DAY
           C
1419         9900 DO 9200 IFG=1,3                                              MOC90000
1420              GO TO (9001,9001,9006),IFG                                   MOC90100
1421         9001 WRITE (6,9002) IFG,K1                                        MOC90200
1422         9002 FORMAT(1H1,48X,15HFOREGROUND AREA,1X,I1,1X,16HSCHEDULE FOR DAY,I2)MOC90300
1423              WRITE (6,9003)                                               MOC90400
1424         9003 FORMAT(1H0,11HTIME- TIME-,27X,5HSETUP,4X,3HCPU)              MOC90500
1425              WRITE (6,9004)                                               MOC90600
1426         9004 FORMAT(1H ,1X,2HON,4X,3HOFF,28X,5H& RWD,1X,6(14HWORK WAIT      ))  MOC90700
1427              WRITE (6,9005)                                               MOC90800
1428              WRITE (6,9039) HON(JS),MON(JS),HOFF(JS),MOFF(JS),PROGNM(IFG),  MOC98050
                 1JOBNAM(JS),STEPNM(JS),ISETUP,IPWND,JWORKH,JWORKM,(PRTCUU(NN),  MOC98060
                 2IWORKH(NN),IWORKM(NN),IWAITH(NN),IWAITM(NN),NN=1,5)          MOC98070
1429         9039 FORMAT(1H0,4(I2,1X),3(A8,1X),2I2,1X,2(2I2,1X),5(A3,1X,2I2,1X,2I2,1MOC98080
                 1X))                                                          MOC98090
1430              GO TO 9049                                                   MOC98110
1431         9040 WRITE (6,9041) (PRTCUU(NN),IWORKH(NN),IWORKM(NN),IWAITH(NN),  MOC98120
                 1IWAITM(NN),NN=1,5)                                           MOC98125
1432         9005 FORMAT(1H ,54HHH MM HH MM PROGRAM  JOB-NAME STEPNAME MM M HHMM HHMMOC90900
                 1M ,5(14HCUU HHMM HHMM ))                                     MOC91000
1433              GO TO 9011                                                   MOC91100
1434         9006 WRITE (6,9007) K1                                            MOC91200
```

```
      9007 FORMAT(1H1,49X,32HBACKGROUND AREA SCHEDULE FOR DAY,I2)           M0091300
           WRITE (6,9008)                                                   M0091400
      9008 FORMAT(1H0,11HTIME- TIME-,18X,5HSETUP,4X,3HCPU)                  M0091500
           WRITE (6,9009)                                                   M0091600
      9009 FORMAT(1H ,1X,2HON,4X,3HOFF,19X,5HE RWD,1X,7(14HWORK WAIT     ))  M0091700
           WRITE (6,9010)                                                   M0091800
      9010 FORMAT(1H ,45HHH MM HH MM JOB-NAME STEPNAME MM M HHMM HHMM ,6(14HM0091900
          1UU HHMM HHMM ))                                                  M0092000
      9011 LS=1                                                             M0092100
           DO 9100 IS=1,MJOB                                                M0092200
           KS=1                                                             M0092300
           JS=ISCHED(IS,1)                                                  M0092400
           IK=JCASE(JS)                                                     M0092450
           IF(JS-1)9100,9012,9012                                           M0092500
      9012 GO TO (9013,9014,9015),IFG                                       M0092600
      9013 GO TO (9016,9100,9100,9016,9100,9016),IK                         M0092700
      9014 GO TO (9100,9016,9100,9100,9016,9016),IK                         M0092800
      9015 GO TO (9100,9100,9016,9016,9016,9016),IK                         M0092900
      9016 MS=SCHED(IFG,JS,4)                                               M0093000
           GO TO (9017,9018),MS                                             M0093100
      9018 GO TO (9019,9017),LS                                             M0093200
      9019 WRITE (6,9020)                                                   M0093300
      9020 FORMAT(66H0*****THE FOLLOWING PROGRAMS COULD NOT BE INCLUDED IN THM0093400
          1E SCHEDULE)                                                      M0093500
           LS=2                                                             M0093600
      9017 HON(JS)=SCHED(IFG,JS,1)/(3600.*1000000.)                         M0093700
           SCHED(IFG,JS,1)=AMOD(SCHED(IFG,JS,1),3600.*1000000)              M0093800
           MON(JS)=(SCHED(IFG,JS,1)+30.)/60.                                M0093900
           HOFF(JS)=SCHED(IFG,JS,2)/(3600.*1000000.)                        M0094000
           SCHED(IFG,JS,2)=AMOD(SCHED(IFG,JS,2),3600.*1000000.)             M0094100
           MOFF(JS)=(SCHED(IFG,JS,2)+30.)/60.                               M0094200
           MOFF(JS)=SCHED(IFG,JS,2)/(60.*1000000.)                          M0094300
           MS=SCHED(IFG,JS,3)                                               M0094400
           IF(TSETUP(IFG,JS,MS))9023,9023,9022                              M0094500
      9022 ISETUP=(TSETUP(IFG,JS,MS)+(30.*1000000.))/(60.*1000000.)         M0094600
           ISETUP=MAX0(1,ISETUP)                                            M0094700
           GO TO 9024                                                       M0094800
      9023 ISETUP=0                                                         M0094900
      9024 IF(TWNDUP(IFG,JS,MS))9026,9026,9025                              M0095000
      9025 IRWND=(TWNDUP(IFG,JS,MS)+(30.*1000000))/(60.*1000000.)           M0095100
           IRWND=MAX0(1,IRWND)                                              M0095200
           GO TO 9027                                                       M0095300
      9026 IRWND=0                                                          M0095400
      9027 JWORKH=TCPU(IFG,JS,MS)/(3600.*1000000.)                          M0095500
           TCPU(IFG,JS,MS)=AMOD(TCPU(IFG,JS,MS),3600.*1000000.)             M0095600
           JWORKM=(TCPU(IFG,JS,MS)+30.)/60.                                 M0095700
           GO TO (9028,9029),LS                                             M0095800
      9028 JWAITH=(60*HOFF(JS)+MOFF(JS)-(60*HON(JS)+MON(JS))-(60*JWORKH+   M0095900
```

```
                    1JWORKM))/60                                                MOC96000
 1481                JWAITM=MOD(60*HOFF(JS)+MOFF(JS)-(60*HON(JS)+MON(JS))-(60*IWORKH+  MOC96100
                    1JWORKM)+30,60)                                             MOC96200
 1482          9029 MM=1                                                        MOC96300
 1483                DO 9030 NS=1,MCUU                                          MOC96400
 1484                PRTCUU(MM)=CUU(NS)                                         MOC96500
 1485                GO TO (9031,9033),MS                                       MOC96600
 1486          9031 IF(TDEV(IFG,JS,NS))9030,9030,9032                           MOC96700
 1487          9032 IWORKH(MM)=TDEV(IFG,JS,NS)/(3600.*1000000.)                 MOC96800
 1488                TDEV(IFG,JS,NS)=AMOD(TDEV(IFG,JS,NS),3600.*1000000.)       MOC96900
 1489                IWORKM(MM)=(TDEV(IFG,JS,NS)+30.)/60.                       MOC97000
 1490                GO TO 9035                                                 MOC97100
 1491          9033 IF(SDEV(IFG,JS,NS))9030,9030,9034                           MOC97200
 1492          9034 IWORKH(MM)=SDEV(IFG,JS,NS)/(3600.*1000000.)                 MOC97300
 1493                SDEV(IFG,JS,NS)=AMOD(SDEV(IFG,JS,NS),3600.*1000000.)       MOC97400
 1494                IWORKM(MM)=(SDEV(IFG,JS,NS)+30.)/60.                       MOC97500
 1495          9035 IWAITH(MM)=(60*HOFF(JS)+MOFF(JS)-(60*HON(JS)+MON(JS))-      MOC97600
                    1(60*IWORKH(MM)+IWORKM(MM)))/60                             MOC97700
 1496                IWAITM(MM)=MOD(60*HOFF(JS)+MOFF(JS)-(60*HON(JS)+MON(JS))-  MOC97800
                    1(60*IWORKH(MM)+IWORKM(MM))+30,60)                          MOC97900
 1497                MM=MM+1                                                    MOC98000
 1498                GO TO (9036,9036,9042),IFG                                 MOC98010
 1499          9036 IF(MM-5)9030,9030,9037                                      MOC98020
 1500          9037 GO TO (9038,9040),KS                                        MOC98030
 1501          9038 KS=2                                                        MOC98040
 1502          9041 FORMAT(1H0,54X,5(A3,1X,2I2,1X,2I2,1X))                      MOC98130
 1503                GO TO 9049                                                 MOC98140
 1504          9042 IF(MM-6)9030,9030,9043                                      MOC98150
 1505          9043 GO TO (9044,9046),KS                                        MOC98160
 1506          9044 KS=2                                                        MOC98170
 1507                WRITE (6,9045) HON(JS),MON(JS),HOFF(JS),MOFF(JS),JOBNAM(JS),  MOC98180
                    1STEPNM(JS),ISETUP,IRWND,JWORKH,JWORKM,(PRTCUU(NN),IWORKH(NN),  MOC98190
                    2IWORKM(NN),IWAITH(NN),IWAITM(NN),NN=1,6)                   MOC98200
 1508          9045 FORMAT(1H0,4(I2,1X),2(A8,1X),2I2,1X,2(2I2,1X),6(A3,1X,2I2,1X,2I2,1MOC98210
                    1X))                                                        MOC98220
 1509          9046 WRITE (6,9047) (PRTCUU(NN),IWORKH(NN),IWORKM(NN),IWAITH(NN),  MOC98230
                    1IWAITM(NN),NN=1,6)                                         MOC98235
 1510          9047 FORMAT(1H0,45X,6(A3,1X,2I2,1X,2I2,1X))                      MOC98240
 1511          9049 MM=1                                                        MOC98250
 1512          9030 CONTINUE                                                    MOC98260
 1513                IF(MM-1)9100,9100,9050                                     MOC98270
 1514          9050 MM=MM-1                                                     MOC98280
 1515                GO TO (9051,9051,9052),IFG                                 MOC98290
 1516          9051 WRITE (6,9041) (PRTCUU(NN),IWORKH(NN),IWORKM(NN),NN=1,MM)   MOC98300
 1517                GO TO 9100                                                 MOC98400
 1518          9052 WRITE (6,9047) (PRTCUU(NN),IWORKH(NN),IWORKM(NN),NN=1,MM)   MOC98500
 1519          9100 CONTINUE                                                    MOC98700
 1520          9200 CONTINUE                                                    MOC98800
```

```
      9999 DO 9996 IS=1,3                                               M0098810
           DO 9996 JS=1,MJOB                                            M0098820
           DO 9996 KS=1,2                                               M0098830
           TCPU(IS,JS,KS)=0.                                            M0098840
           TNOVLP(IS,JS,KS)=0.                                          M0098850
           TMPXBA(IS,JS,KS)=0.                                          M0098860
           TMPXBW(IS,JS,KS)=0.                                          M0098870
           TSEL1A(IS,JS,KS)=0.                                          M0098880
           TSEL1W(IS,JS,KS)=0.                                          M0098890
           TSEL2A(IS,JS,KS)=0.                                          M0098900
           TSEL2W(IS,JS,KS)=0.                                          M0098910
           TSEL3A(IS,JS,KS)=0.                                          M0098920
           TSEL3W(IS,JS,KS)=0.                                          M0098930
           TSEL4A(IS,JS,KS)=0.                                          M0098940
           TSEL4W(IS,JS,KS)=0.                                          M0098950
           TSEL5A(IS,JS,KS)=0.                                          M0098960
           TSEL5W(IS,JS,KS)=0.                                          M0098970
           TSEL6A(IS,JS,KS)=0.                                          M0098980
           TSEL6W(IS,JS,KS)=0.                                          M0098990
           TSETUP(IS,JS,KS)=0.                                          M0099000
           TWNDUP(IS,JS,KS)=0.                                          M0099010
           TMPXMA(IS,JS,KS)=0.                                          M0099020
           TMPXMW(IS,JS,KS)=0.                                          M0099030
      9996 TOTAL(IS,JS,KS)=0.                                           M0099040
           DO 9995 IS=1,3                                               M0099050
           DO 9995 JS=1,MJOB                                            M0099060
           DO 9995 KS=1,MCUU                                            M0099070
           TDEV(IS,JS,KS)=0.                                            M0099080
      9995 SDEV(IS,JS,KS)=0.                                            M0099090
           I1=IJ                                                        M0099100
           GO TO 870                                                    M0099110
C ****************************************************************************
C
C ERROR MESSAGES
C
C ****************************************************************************
      2000 FORMAT(18H PROGRAM EXPECTED ,A1,11H IN COLUMN ,A1)           M0100100
      2001 FORMAT(18H PROGRAM EXPECTED ,A1,1X,2HOR,1X,A1,11H IN COLUMN ,A1) M0100200
      1002 CALL ERRPRT (ERROR)                                          M0100300
           WRITE (6,2000) H,D1                                          M0100400
      2002 IFIXIT=10*(AVINST+.05)                                       M0100420
           JFIXIT=100*(AVSEL+.005)                                      M0100440
           KFIXIT=10*(AVMPX+.05)                                        M0100460
           WRITE (6,3002) A1,A2,CPU,IFIXIT,JFIXIT,KFIXIT                M0100500
      3002 FORMAT(1H ,2A1,I4,3I3)                                       M0100510
           WRITE (6,4050)                                               M0100520
      4050 FORMAT(38H0HC STATEMENT INVALID.  RUN CANCELLED.)            M0100530
           WRITE (6,4051)                                               M0100540
```

```
1564       4051 FORMAT(59HOVERIFY THAT DICTIONARY STATEMENT IS FIRST STATEMENT IN M0100550
               1INPUT STREAM.)                                                    M0100560
1565            WRITE (6,4052)                                                    M0100570
1566       4052 FORMAT(39HOCORRECT HC STATEMENT AND RESUBMIT RUN.)                M0100580
1567            CALL EXIT                                                         M0100600
1568       1003 CALL ERRPRT (ERROR)                                               M0100700
1569            WRITE (6,2000) C,D2                                               M0100800
1570            GO TO 2002        (CALL ERRPRT (ERROR)                            M0100900
1571       1330 WRITE (6,2330)                                                    M0100930
1572       2330 FORMAT(72H CPU MODEL NUMBER IN COLUMNS 3-6 MUST BE 2030, 2040, 205M0100950
               10, 2065, OR 2075)                                                 M0100970
1573            GO TO 2002                                                        M0100980
1574       1005 CALL ERRPRT (ERROR)                                               M0101000
1575            WRITE (6,2000) 4,D1                                               M0101100
1576       2005 IFIXIT=DEVRCD(1)                                                  M0101110
1577            JFIXIT=DEVSPD(1)                                                  M0101120
1578            KFIXIT=10*(DEVSTP(1)+.05)                                         M0101130
1579            LFIXIT=DEVLSS(1)                                                  M0101140
1580            MFIXIT=DEVSRW(1)                                                  M0101150
1581            NFIXIT=DEVDEN(1)                                                  M0101160
1582            OFIXIT=10*(DEVINT(1)+.05)                                         M0101170
1583            WRITE (6,3005) A1,A2,DEVTYP(1),DEVMOD(1),DEVCLS(1),IFIXIT,JFIXIT, M0101200
               1KFIXIT,LFIXIT,MFIXIT,NFIXIT,OFIXIT                                M0101300
1584       3005 FORMAT(1H ,2A1,A4,A2,A1,I3,I4,I2,4I3)                             M0101350
1585            GO TO 4                                                           M0101400
1586       1006 CALL ERRPRT (ERROR)                                               M0101500
1587            WRITE (6,2000) D,D2                                               M0101600
1588            GO TO 2005                                                        M0101700
1589       1009 CALL ERRPRT (ERROR)                                               M0101800
1590            WRITE (6,2009)                                                    M0101900
1591       2009 FORMAT(44H DEVICE CLASS IN COLUMN 9 MUST BE D, T, OR U)           M0102000
1592            GO TO 2005                                                        M0102100
1593       1010 CALL ERRPRT (ERROR)                                               M0102200
1594            WRITE (6,2010)                                                    M0102300
1595       2010 FORMAT(55H INTEFERENCE TIME IN COLUMNS 25-27 MUST BE NON-NEGATIVE)M0102400
1596            GO TO 2005                                                        M0102500
1597       1012 CALL ERRPRT (ERROR)                                               M0102600
1598            WRITE (6,2000) H,D1                                               M0102700
1599       2012 IFIXIT=DEVRCD(I1)                                                 M0102710
1600            JFIXIT=DEVSPD(I1)                                                 M0102720
1601            KFIXIT=10*(DEVSTP(I1)+.05)                                        M0102730
1602            LFIXIT=DEVLSS(I1)                                                 M0102740
1603            MFIXIT=DEVSRW(I1)                                                 M0102750
1604            NFIXIT=DEVDEN(I1)                                                 M0102760
1605            OFIXIT=10*(DEVINT(I1)+.05)                                        M0102770
1606            WRITE (6,3005) A1,A2,DEVTYP(I1),DEVMOD(I1),DEVCLS(I1),IFIXIT,     M0102800
               1JFIXIT,KFIXIT,LFIXIT,MFIXIT,NFIXIT,OFIXIT                         M0102900
1607            IDEV=I1                                                           M0103000
```

```
1608           GO TO 12                                                     M0103100
1609      1014 CALL ERRPRT (ERROR)                                          M0103200
1610           WRITE (6,2001) D,Z,D2                                        M0103300
1611           GO TO 2012                                                   M0103400
1612      1017 CALL ERRPRT (ERROR)                                          M0103500
1613           WRITE (6,2009)                                               M0103600
1614           GO TO 2012                                                   M0103700
1615      1019 CALL ERRPRT (ERROR)                                          M0103800
1616           WRITE (6,2010)                                               M0103900
1617           GO TO 2012                                                   M0104000
1618      1020 CALL ERRPRT (ERROR)                                          M0104100
1619           WRITE (6,2020) DEVTYP(I1),DEVMOD(I1)                         M0104200
1620      2020 FORMAT(40H DUPLICATE DEVICE TYPE AND MODEL NUMBER ,A4,1X,A2) M0104300
1621           GO TO 2012                                                   M0104400
1622      1211 CALL ERRPRT (ERROR)                                          M0104500
1623           WRITE (6,2000) H,D1                                          M0104600
1624      2211 WRITE (6,3211) A1,A2                                         M0104700
1625      3211 FORMAT(1H ,2A1)                                              M0104750
1626           GO TO 210                                                    M0104800
1627      1212 CALL ERRPRT (ERROR)                                          M0104900
1628           WRITE (6,2000) Z,D2                                          M0105000
1629           GO TO 2211                                                   M0105100
1630      1024 CALL ERRPRT (ERROR)                                          M0105200
1631           WRITE (6,2000) I,D1                                          M0105300
1632      2024 WRITE (6,3023) A1,A2,A3,A4,N1,(PRTCUU(K),K=1,23),A5          M0105400
1633      3023 FORMAT(1H ,2A1,A4,A2,I2,23A3,A1)                             M0105450
1634           N1=0                                                         M0105500
1635           K=I1-1                                                       M0105600
1636           GO TO 303                                                    M0105700
1637      1240 CALL ERRPRT (ERROR)                                          M0105800
1638           WRITE (6,2000) D,D2                                          M0105900
1639           GO TO 2024                                                   M0106000
1640      1241 CALL ERRPRT (ERROR)                                          M0106100
1641           IP1=I1-1                                                     M0106200
1642           WRITE (6,2241) N1,IP1,MCUU                                   M0106300
1643      2241 FORMAT(31H NUMBER OF UNITS IN STATEMENT (,I2,43H) PLUS NUMBER OF UM0106400
              1NITS PREVIOUSLY DEFINED (,I2,27H) EXCEEDS PROGRAM MAXIMUM (,I2,1H)M0106500
              2)                                                            M0106600
1644           GO TO 2024                                                   M0106700
1645      1025 CALL ERRPRT (ERROR)                                          M0106800
1646           WRITE (6,2025) N1                                            M0106900
1647      2025 FORMAT(18H NUMBER OF UNITS (,I2,39H) IN COLUMNS 9-10 EXCEEDS MAXIMM0107000
              1UM OF 23)                                                    M0107100
1648           GO TO 2024                                                   M0107200
1649      1026 CALL ERRPRT (ERROR)                                          M0107220
1650           WRITE (6,2026)                                               M0107230
1651      2026 FORMAT(31H NUMBER OF UNITS IS LESS THAN 1)                   M0107240
1652           GO TO 2024                                                   M0107260
```

```
1653       1028 CALL ERRPRT (ERROR)                                          M0107300
1654            WRITE (3,2028) A3,A4                                         M0107400
1655       2028 FORMAT(30H DEVICE TYPE AND MODEL NUMBER ,A4,1X,A2,1X,30HNOT DEFINEM0107500
               1D BY AN HD STATEMENT)                                        M0107600
1656            GO TO 2024                                                   M0107700
1657       1029 CALL ERRPRT (ERROR)                                          M0107800
1658            WRITE (6,2029) CUU(K)                                        M0107900
1659       2029 FORMAT(23H DUPLICATE UNIT NUMBER ,A3)                        M0108000
1660            GO TO 2024                                                   M0108100
1661       1300 CALL ERRPRT (ERROR)                                          M0108900
1662            PRINT 2300,CUU(K)                                            M0108900
1663       2300 FORMAT(24H INVALID CHANNEL NUMBER ,A3)                       M0109000
1664            GO TO 2024                                                   M0109100
1665       1033 CALL ERRPRT (ERROR)                                          M0109200
1666            PRINT 2000,I,D1                                              M0109300
1667       2033 IFIXIT=FINSTX(IFG)                                           M0109310
1668            JFIXIT=FBLKFC(IFG)                                           M0109320
1669            KFIXIT=FLTNCY(IFG)                                           M0109330
1670            LFIXIT=FSEEK(IFG)                                            M0109340
1671            WRITE (6,3033) A1,A2,N1,A3,A4,PROGNM(IFG),FGCUUP(IFG),FGCUUI(IFG),M0109400
               1IFIXIT,JFIXIT,KFIXIT,LFIXIT                                  M0109500
1672       3033 FORMAT(1H ,2A1,I1,2A1,A8,2A3,I3,I2,2I3)                      M0109600
1673            I2=IFG                                                       M0109700
1674            GO TO 320                                                    M0109800
1675       1034 CALL ERRPRT (ERROR)                                          M0109900
1676            WRITE (6,2000) F,D2                                          M0110000
1677            GO TO 2033                                                   M0110100
1678       1035 CALL ERRPRT (ERROR)                                          M0110200
1679            GO TO (2034,2035),IFG                                        M0110300
1680       2034 WRITE (6,2000) D1,D3                                         M0110400
1681            GO TO 2033                                                   M0110500
1682       2035 WRITE (6,2000) D2,D3                                         M0110600
1683            GO TO 2033                                                   M0110700
1684       1037 CALL ERRPRT (ERROR)                                          M0110800
1685            WRITE (6,2037)                                               M0110900
1686       2037 FORMAT(35H PROGRAM EXPECTED BLANK IN COLUMN 5)               M0111000
1687       2038 IFIXIT=FINSTX(IFG)                                           M0111010
1688            JFIXIT=FBLKFC(IFG)                                           M0111020
1689            KFIXIT=FLTNCY(IFG)                                           M0111030
1690            LFIXIT=FSEEK(IFG)                                            M0111040
1691            WRITE (6,3033) A1,A2,N1,A3,A4,PROGNM(IFG),FGCUUP(IFG),FGCUUI(IFG),M0111100
               1IFIXIT,JFIXIT,KFIXIT,LFIXIT                                  M0111200
1692            FG(IFG)=BLANK4                                               M0111300
1693            GO TO (2039,680),IFG                                         M0111400
1694       2039 I2=2                                                         M0111500
1695            GO TO 320                                                    M0111600
1696       1044 CALL ERRPRT (ERROR)                                          M0111700
1697            WRITE (6,2044)                                               M0111800
```

```
1698    2044 FORMAT(47H PROGRAM EXPECTED P, R, T, OR BLANK IN COLUMN 4)     MO111900
1699         GO TO 2038                                                     MO112000
1700    1046 CALL ERRPRT (ERROR)                                            MO112100
1701         WRITE (6,2001) I,O,D5                                          MO112200
1702         GO TO 2038                                                     MO112300
1703    1049 CALL ERRPRT (ERROR)                                            MO112400
1704         WRITE (6,2049)                                                 MO112500
1705    2049 FORMAT(43H NO FOREGROUND PROGRAM NAME IN COLUMNS 6-13)         MO112600
1706         GO TO 2038                                                     MO112700
1707    1050 CALL ERRPRT (ERROR)                                            MO112800
1708         WRITE (6,2050)                                                 MO112900
1709    2050 FORMAT(48H DEVICE CLASS OF UNIT IN COLUMNS 14-15 MUST BE U)    MO113000
1710         GO TO 2038                                                     MO113100
1711    1051 CALL ERRPRT (ERROR)                                            MO113200
1712         WRITE (6,2051) FGCUUP(IFG)                                     MO113300
1713    2051 FORMAT(13H UNIT NUMBER ,A3,31H NOT DEFINED BY AN ID STATEMENT) MO113400
1714         GO TO 2038                                                     MO113500
1715    1054 CALL ERRPRT (ERROR)                                            MO113600
1716         WRITE (6,2054)                                                 MO113700
1717    2054 FORMAT(53H DEVICE CLASS OF UNIT IN COLUMNS 17-19 MUST BE D OR T) MO113800
1718         GO TO 2038                                                     MO113900
1719    1055 CALL ERRPRT (ERROR)                                            MO114000
1720         WRITE (6,2051) FGCUUI(IFG)                                     MO114100
1721         GO TO 2038                                                     MO114200
1722    1056 CALL ERRPRT (ERROR)                                            MO114300
1723         WRITE (6,2056)                                                 MO114400
1724    2056 FORMAT(51H LATENCY TIME IN COLUMNS 25-27 MUST BE NON-NEGATIVE) MO114500
1725         GO TO 2038                                                     MO114600
1726    1058 CALL ERRPRT (ERROR)                                            MO114700
1727         WRITE (6,2058)                                                 MO114800
1728    2058 FORMAT(48H SEEK TIME IN COLUMNS 28-30 MUST BE NON-NEGATIVE)    MO114900
1729         GO TO 2038                                                     MO115000
1730    1066 CALL ERRPRT (ERROR)                                            MO115100
1731         WRITE (6,2066) A0                                              MO115200
1732    2066 FORMAT(40H BOTH FOREGROUND PROGRAMS ASSIGNED UNIT ,A3)         MO115300
1733         FG(1)=BLANK4                                                   MO115330
1734         FG(2)=BLANK4                                                   MO115360
1735         GO TO 680                                                      MO115400
1736    1074 CALL ERRPRT (ERROR)                                            MO116700
1737         WRITE (6,2000) I,D1                                            MO116800
1738    2074 WRITE (6,3074) A1,A2,NI,(HHON(I1),MMON(I1),HHOFF(I1),MMOFF(I1), MO116850
            1I1=1,7)                                                        MO116900
1739    3074 FORMAT(1H ,2A1,I1,7(4I2))                                      MO116950
1740         GO TO 730                                                      MO117000
1741    1075 CALL ERRPRT (ERROR)                                            MO117100
1742         WRITE (6,2000) S,D2                                            MO117200
1743         GO TO 2074                                                     MO117300
1744    1750 CALL ERRPRT (ERROR)                                            MO117400
```

```
1745          WRITE (6,2750)                                                     MO117500
1746     2750 FORMAT(41H NUMBER OF DAYS IN WEEK IS GREATER THAN 7)               MO117600
1747          WEEK=7                                                             MO117700
1748     2752 WRITE (6,3074) A1,A2,N1,(HHON(I1),MMON(I1),HHOFF(I1),MMOFF(I1),    MO117800
             1I1=1,7)                                                            MO117850
1749          GO TO 76                                                           MO117900
1750     1751 CALL ERRPRT (ERROR)                                                MO117920
1751          WRITE (6,2751)                                                     MO117940
1752     2751 FORMAT(38H NUMBER OF DAYS IN WEEK IS LESS THAN 1)                  MO117960
1753          WEEK=1                                                             MO117970
1754          GO TO 2752                                                         MO117980
1755     1760 CALL ERRPRT (ERROR)                                                MO118000
1756          WRITE (6,2760) I1                                                  MO118100
1757     2760 FORMAT(17H HOUR ON FOR DAY ,I1,19H IS GREATER THAN 23)             MO118200
1758          HHON(I1)=23                                                        MO118250
1759     3760 WRITE (6,3074) A1,A2,N1,(HHON(J1),MMON(J1),HHOFF(J1),MMOFF(J1),    MO118300
             1J1=1,7)                                                            MO118350
1760          GO TO 77                                                           MO118400
1761     1770 CALL ERRPRT (ERROR)                                                MO118420
1762          WRITE (6,2770) I1                                                  MO118440
1763     2770 FORMAT(17H HOUR ON FOR DAY ,I1,15H IS LESS THAN 0)                 MO118460
1764          HHON(I1)=0                                                         MO118470
1765          GO TO 3760                                                         MO118480
1766     1077 CALL ERRPRT (ERROR)                                                MO118500
1767          WRITE (6,2077) I1                                                  MO118600
1768     2077 FORMAT(18H HOUR OFF FOR DAY ,I1,19H IS GREATER THAN 23)            MO118700
1769          HHOFF(I1)=23                                                       MO118750
1770     3077 WRITE (6,3074) A1,A2,N1,(HHON(J1),MMON(J1),HHOFF(J1),MMOFF(J1),    MO118800
             1J1=1,7)                                                            MO118850
1771          GO TO 78                                                           MO118900
1772     1780 CALL ERRPRT (ERROR)                                                MO118920
1773          WRITE (6,2780) I1                                                  MO118940
1774     2780 FORMAT(18H HOUR OFF FOR DAY ,I1,15H IS LESS THAN 0)                MO118960
1775          HHOFF(I1)=0                                                        MO118970
1776          GO TO 3077                                                         MO118980
1777     1078 CALL ERRPRT (ERROR)                                                MO119000
1778          WRITE (6,2078) I1                                                  MO119100
1779     2078 FORMAT(19H MINUTE ON FOR DAY ,I1,19H IS GREATER THAN 59)           MO119200
1780          MMON(I1)=59                                                        MO119250
1781     3078 WRITE (6,3074) A1,A2,N1,(HHON(J1),MMON(J1),HHOFF(J1),MMOFF(J1),    MO119300
             1J1=1,7)                                                            MO119350
1782          GO TO 79                                                           MO119400
1783     1790 CALL ERRPRT (ERROR)                                                MO119420
1784          WRITE (6,2790) I1                                                  MO119440
1785     2790 FORMAT(19H MINUTE ON FOR DAY ,I1,15H IS LESS THAN 0)               MO119460
1786          MMON(I1)=0                                                         MO119470
1787          GO TO 3078                                                         MO119480
1788     1079 CALL ERRPRT (ERROR)                                                MO119500
```

```
1789           WRITE (6,2079) I1                                                  M0119600
1790      2079 FORMAT(20H MINUTE OFF FOR DAY ,I1,19H IS GREATER THAN 59)          M0119700
1791           MMOFF(I1)=59                                                       M0119750
1792      3079 WRITE (6,3074) A1,A2,N1,(HHON(J1),MMON(J1),HHOFF(J1),MMOFF(J1),    M0119800
              1J1=1,7)                                                            M0119850
1793           GO TO 80                                                           M0119900
1794      1800 CALL ERRPRT (ERROR)                                                M0119920
1795           WRITE (6,2800) I1                                                  M0119940
1796      2800 FORMAT(20H MINUTE OFF FOR DAY ,I1,15H IS LESS THAN 0)              M0119960
1797           MMOFF(I1)=0                                                        M0119970
1798           GO TO 3079                                                         M0119980
1799      1081 CALL ERRPRT (ERROR)                                                M0120500
1800           WRITE (6,2081) I1                                                  M0120600
1801      2081 FORMAT(18H TIME OFF FOR DAY ,I1,33H IS LESS THAN OR EQUAL TO TIME M0120700
              1ON)                                                                M0120800
1802           WRITE (6,3074) A1,A2,N1,(HHON(J1),MMON(J1),HHOFF(J1),MMOFF(J1),    M0120900
              1J1=1,7)                                                            M0120950
1803           GO TO 82                                                           M0121000
1804      1085 CALL ERRPRT (ERROR)                                                M0121100
1805           WRITE (6,2000) I,D1                                                M0121200
1806      2085 WRITE (6,3085) A1,A2                                               M0121300
1807      3085 FORMAT(1H ,2A1)                                                    M0121350
1808           GO TO 850                                                          M0121400
1809      1085 CALL ERRPRT (ERROR)                                                M0121500
1810           WRITE (6,2000) Z,D2                                                M0121600
1811           GO TO 2085                                                         M0121700
1812      1089 CALL ERRPRT (ERROR)                                                M0121800
1813           WRITE (6,2000) J,D1                                                M0121900
1814      2089 IFIXIT=FIXINS                                                      M0121910
1815           JFIXIT=VARINS                                                      M0121920
1816           KFIXIT=100*(NOVLP+.005)                                            M0121930
1817           WRITE (6,3089) A1,A2,A3,JOBNAM(IJ),STEPNM(IJ),DON(IJ),HON(IJ),     M0122000
              1MON(IJ),DOFF(IJ),HOFF(IJ),MOFF(IJ),IFIXIT,JFIXIT,PRECED,KFIXIT     M0122050
1818      3089 FORMAT(1H ,3A1,2X,2A8,I1,2I2,I1,2I2,2I5,A8,I2)                     M0122100
1819      2090 I1=IJ                                                              M0122200
1820           GO TO 870                                                          M0122300
1821      1091 CALL ERRPRT (ERROR)                                                M0122400
1822           WRITE (6,2000) DAY,D2                                              M0122500
1823           GO TO 2089                                                         M0122600
1824      1092 ERRORX=0                                                           M0122630
1825           GO TO 2089                                                         M0122660
1826      1093 CALL ERRPRT (ERROR)                                                M0122700
1827           WRITE (6,2093)                                                     M0122800
1828      2093 FORMAT(48H TIME ON AND TIME OFF DO NOT AGREE WITH JOB TYPE)        M0122900
1829           GO TO 2089                                                         M0123100
1830      1940 CALL ERRPRT (ERROR)                                                M0123200
1831           WRITE (6,2940)                                                     M0123300
1832      2940 FORMAT(56H TIME ON AND TIME OFF INDICATE STATEMENT OUT OF SEQUENCEM0123400
```

```
     1)                                                                 M0123500
      GO TO 2089                                                        M0123700
 1096 CALL ERRPRT (ERROR)                                               M0123800
      WRITE (6,2000) F,D3                                               M0123900
      GO TO 2089                                                        M0124000
 1097 CALL ERRPRT (ERROR)                                               M0124100
      WRITE (6,2001) B,F,D3                                             M0124200
      GO TO 2089                                                        M0124300
 1910 CALL ERRPRT (ERROR)                                               M0124400
      WRITE (6,2910)                                                    M0124500
 2910 FORMAT(31H DAY IN COLUMN 2 GREATER THAN 7)                        M0124600
      GO TO 2089                                                        M0124700
 1911 CALL ERRPRT (ERROR)                                               M0124800
      WRITE (6,2911)                                                    M0124900
 2911 FORMAT(26H DAY IN COLUMN 2 NOT VALID)                             M0125000
      I5=I3                                                             M0125050
      GO TO 2089                                                        M0125100
 1971 CALL ERRPRT (ERROR)                                               M0125200
      WRITE (6,2971)                                                    M0125300
 2971 FORMAT(90H STEPNAME IN COLUMNS 42-49 OF THIS STATEMENT DOES NOT AGM0125400
     1REE WITH STEPNAME OF PREVIOUS STEP)                               M0125500
      GO TO 2089                                                        M0125700
 1972 CALL ERRPRT (ERROR)                                               M0125800
      WRITE (6,2972)                                                    M0125900
 2972 FORMAT(95H TIME ON OR TIME OFF IN THIS STATEMENT DOES NOT AGREE WIM0126000
     1TH TIME ON OR TIME OFF OF PREVIOUS STEP)                          M0126100
      GO TO 2089                                                        M0126300
 1978 CALL ERRPRT (ERROR)                                               M0126400
      WRITE (6,2978) JOBNAM(IJ)                                         M0126500
 2978 FORMAT(20H DUPLICATE JOB NAME ,A8)                                M0126600
      GO TO 2089                                                        M0126800
 1099 CALL ERRPRT (ERROR)                                               M0126900
      WRITE (6,2000) J,D1                                               M0127000
 2099 WRITE (6,3099) A0,A4,A5,A6,A7,A8,FILEFG(IJ,IFG)                   M0127100
 3099 FORMAT(1H ,4A1,1X,2A1,A7)                                         M0127150
      I7=3                                                              M0127200
      I8=IFG                                                            M0127300
      GO TO 2090                                                        M0127400
 1990 CALL ERRPRT (ERROR)                                               M0127500
      WRITE (6,2000) DAY,D2                                             M0127600
      GO TO 2099                                                        M0127700
 1101 CALL ERRPRT (ERROR)                                               M0127800
      WRITE (6,2000) B,D3                                               M0127900
      GO TO 2099                                                        M0128000
 1102 CALL ERRPRT (ERROR)                                               M0128100
      WRITE (6,2000) F,D4                                               M0128200
      GO TO 2099                                                        M0128300
 1104 CALL ERRPRT (ERROR)                                               M0128400
```

```
      WRITE (6,2104)                                                    M0128500
 2104 FORMAT(35H PROGRAM EXPECTED BLANK IN COLUMN 7)                    M0128600
 2112 WRITE (6,3099) A0,A4,A5,A6,A7,A8,FILEFG(IJ,IFG)                   M0128700
      FLFGPT(IJ,IFG)=0                                                  M0128750
      GO TO (2113,2114),IFG                                             M0128800
 2113 I7=3                                                              M0128900
      I8=2                                                              M0129000
      GO TO 2090                                                        M0129100
 2114 I7=4                                                              M0129200
      GO TO 2090                                                        M0129300
 1109 CALL ERRPRT (ERROR)                                               M0129400
      WRITE (6,2109)                                                    M0129500
 2109 FORMAT(47H PROGRAM EXPECTED P, R, T, OR BLANK IN COLUMN 6)        M0129600
      GO TO 2112                                                        M0129700
 1112 CALL ERRPRT (ERROR)                                               M0129800
      WRITE (6,2001) T,0,D7                                             M0129900
      GO TO 2112                                                        M0130000
 1125 CALL ERRPRT (ERROR)                                               M0130100
      WRITE (6,2125)                                                    M0130200
 2125 FORMAT(29H NO FILE NAME IN COLUMNS 8-14)                          M0130300
      GO TO 2112                                                        M0130400
 1126 CALL ERRPRT (ERROR)                                               M0130600
      WRITE (6,2126) FILEFG(IJ,IFG),DAY                                 M0130700
 2126 FORMAT(6H FILE ,A7,19H NOT DEFINED BY A J,A1,12HBD STATEMENT)     M0130800
      GO TO 2112                                                        M0130900
 1127 CALL ERRPRT (ERROR)                                               M0131000
      WRITE (6,2127)                                                    M0131100
 2127 FORMAT(47H SAME FILE ASSIGNED TO BOTH FOREGROUND PROGRAMS)        M0131200
      I7=4                                                              M0131300
      GO TO 2090                                                        M0131400
 1129 CALL ERRPRT (ERROR)                                               M0131410
      WRITE (6,2129)                                                    M0131430
 2129 FORMAT(46H TWO FILES ASSIGNED TO SAME FOREGROUND PROGRAM)         M0131450
      I7=4                                                              M0131470
      GO TO 2090                                                        M0131490
 1135 CALL ERRPRT (ERROR)                                               M0131500
      WRITE (6,2135) FILEFG(IJ,IFG)                                     M0131600
 2135 FORMAT(17H DEVICE CALSS OF ,A7,10H MUST BE 0)                     M0131700
      IF(EIGHTY-BLANK4)2137,2138,2137                                   M0131800
 2137 I7=4                                                              M0131900
      GO TO 2090                                                        M0132000
 2138 I7=5                                                              M0132100
      I8=1                                                              M0132200
      GO TO 2090                                                        M0132300
 1154 CALL ERRPRT (ERROR)                                               M0132900
      WRITE (6,2154) A2,A3,FILNAM(J1)                                   M0133000
 2154 FORMAT(2H J,2A1,30H10 STATEMENT MISSING FOR FILE ,A7)             M0133100
      DO 2155 IP1=1,2                                                   M0133120
```

```
      IF(FILEFG(IJ,IP1)-FILNAM(J1))2155,2156,2155                       M0133140
 2156 FLFGPT(IJ,IP1)=0.                                                 M0133160
 2155 CONTINUE                                                          M0133180
      I7=5                                                              M0133200
      I8=J1+1                                                           M0133300
      GO TO 2090                                                        M0133400
 1193 CALL ERRPRT (ERROR)                                               M0133500
      WRITE (6,2000) J,D1                                               M0133600
 2193 WRITE (6,3193) A1,A2                                              M0133700
 3193 FORMAT(1H ,2A1)                                                   M0133750
      I7=6                                                              M0133800
      GO TO 2090                                                        M0133900
 1194 CALL ERRPRT (ERROR)                                               M0134000
      WRITE (6,2000) Z,D2                                               M0134100
      GO TO 2193                                                        M0134200
      END                                                               M9999999
```

SCALAR MAP

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
BLANK8	18C8	PRECED	18D0	NAME	18D8	MDEV	18E0	MCUU	18E4
MFILE	18E8	MJOB	18EC	MDJOB	18F0	MWJOB	18F4	ERROR	18F8
ERRORX	18FC	I1	1900	J1	1904	K1	1908	A	190C
B	1910	C	1914	D	1918	F	191C	H	1920
I	1924	J	1928	O	192C	P	1930	R	1934
S	1938	T	193C	U	1940	X	1944	Z	1948
D0	194C	D1	1950	D2	1954	D3	1958	D4	195C
D5	1960	D6	1964	D7	1968	BLANK4	196C	A1	1970
A2	1974	CPU	1978	AVINST	197C	AVSEL	1980	AVMPX	1984
IDEV	1988	K	198C	A3	1990	A4	1994	N1	1998
A5	199C	L	19A0	M	19A4	N	19A8	NN	19AC
I2	19B0	IFG	19B4	A0	19B8	CASE	19BC	TIMEE	19C0
TIMEB	19C4	WEEK	19C8	IFTXIT	19CC	FFIXIT	19D0	I1	19D4
I3	19D8	I4	19DC	I5	19E0	I6	19E4	I7	19E8
DAY	19EC	MMJOB	19F0	IJ	19F4	FIXINS	19F8	VARINS	19FC
NOVLP	1A00	TCASE	1A04	MXDJOB	1A08	MMMJOB	1A0C	JJ	1A10
MXJOB	1A14	MXWJOB	1A18	IL	1A1C	IK	1A20	IXI	1A24
I8	1A28	A6	1A2C	A7	1A30	A8	1A34	IPI	1A38
VOLUME	1A3C	RCDSIZ	1A40	BLKFCT	1A44	MULT	1A48	LTNCY	1A4C
SEEK	1A50	FACTOR	1A54	IOPNT	1A58	EIGHTY	1A5C	CLASS	1A60
DEVICE	1A64	K2	1A68	K3	1A6C	IFGX	1A70	IX	1A74
CUUP	1A78	CIPT	1A7C	CUUI	1A80	CPPT	1A84	IX3	1A88
ISTDJ	1A8C	ISIJ	1A90	BEGIN	1A94	END	1A98	IS	1A9C
IS	1AA0	JS	1AA4	KS	1AA8	FRCMB2	1AAC	TIMEC	1AB0
TIMEX	1AB4	NS	1AB8	MS	1ABC	TSETUP	1AC0	IRWND	1AC4
JWORKH	1AC8	JWORKM	1ACC	JWAITH	1AD0	JWAITM	1AD4	MM	1AD8
JFIXIT	1ADC	KFIXIT	1AE0	LFIXIT	1AE4	MFIXIT	1AE8	NFIXIT	1AEC
OFIXIT	1AF0								

ARRAY MAP

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
QL	1AF4	QN	1B08	FGUPPT	1B1C	FGUIPT	1B24	CUUPNT	1B2C
FGCPPT	1B7C	FGCIPT	1B84	FILPNT	1B8C	HHON	1BDC	HHOFF	1BE8
DON	1C14	HON	1C64	DOFF	1CB4	HOFF	1D04	DEVTYP	1D54
DEVMOD	1D7C	DEVCLS	1DA4	DEVSPD	1DCC	DEVRCD	1DF4	DEVSTR	1E1C
DEVLSS	1E44	DEVSRW	1E6C	DEVDEN	1E94	DEVINT	1EBC	CUU	1EE4
FGCUUP	1F34	FGCUUI	1F3C	TOTAL	1F44	TWORKH	2124	TWORKM	2140
IWAITH	215C	IWAITM	2178	BXFRCM	2194	BXFRCB	21F4	BXFRCC	2254
ISCHED	22B4	SCHED	2354	FG	2714	MMON	271C	MMOFF	2738
PRTCUU	2754	MON	27CC	MOFF	281C	TCPU	286C	TNOVLP	2A4C
FLFGPT	2C2C	FLNMPT	2CCC	FLFCTR	2D6C	FLIOPT	2D74	FLDEV	2D7C
FLMULT	2D84	FLVOL	2D8C	TDEV	2D94	SDEV	4054	TMPXBA	5314
TMPXBW	54F4	TSEL1A	56D4	TSEL1W	58B4	TSEL2A	5A94	TSEL2W	5C74
TSEL3A	5E54	TSEL3W	6034	TSEL4A	6214	TSEL4W	63F4	TSEL5A	65D4
TSEL5W	67B4	TSEL6A	6994	TSEL6W	6B74	TSETUP	6D54	TWNDUP	6F34

EXPXMA	7114	TMPXMN	72F4	FINSTX	74D4	FBLKFC	740C	FLENCY	74F4
FSEEK	74EC	JOBNAM	74F8	PROGNM	7598	FILNAM	75A8	STFPNW	7548
FILEFG	76F8	FRCMF1	7828	FRCBF1	7888	FRCCF1	78E8	FRCMF2	7948
FRCBF2	79A8	FRCCF2	7A08	BGFRCM	7A68	BGFRCB	7AC8	BGFRCC	7B28
FIFRCM	7B88	FIFRCB	7BE8	FIFRCC	7C48	F2FRCM	7CA8	F2FRCB	7D08
F2FRCC	7D68	FRACM	7DC8	FRACB	8548	FRACC	8CC8	FRCMAF	9440
FRCMBF	94A0	FRCMCF	94F8	FRCBMF	9550	FRCBBF	95A8	FRCBCF	9600
FRCCMF	9658	FRCCBF	96B0	FRCCCF	9708	FRCMMX	9760	FRCMBX	97B8
FRCMCX	9810	FRCBMX	9868	FRCBBX	98C0	FRCBCX	9918	FRCCMX	9970
FRCCBX	99C8	FRCCCX	9A20	FRACMX	9A78	FRACBX	A1F8	FRACCX	A078
QA	30F8	QB	B10C	QC	B120	QD	B134	QDX	B148
QDP	B15C	QDPX	B170	JCASE	B184				

SUBPROGRAMS CALLED

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
IBCOM=	B1D4	UNITX	B1D8	TAPEX	B1DC	DISKX	B1E0	FRCTN	B1E4
TIME23	B1E8	TOD	B1EC	SWITCH	B1F0	FILE	B1F4	FILEIO	B1F8
EXIT	B1FC	IGNSCH	B200	JOHNSN	B204	ERRPRT	B208	MINO	B20C
AMAX1	B210	MAXO	B214						

FORMAT STATEMENT MAP

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
4004	C3FC	4005	C41E	4006	C43B	4007	C491	4008	C4F7
4009	C545	4010	C585	4011	C5C8	4012	C5FF	4013	C673
1	C6C8	2	C6D2	5	C6E3	211	C700	24	C706
33	C713	74	C735	85	C744	4015	C74A	4016	C774
4017	C790	4018	C7C4	4019	C812	4020	C861	4021	C893
4022	C8A8	4023	C8D3	4024	C917	4025	C94F	4026	C998
4027	C9E2	4028	C9FA	4029	CA2F	4030	CA5F	4031	CA8F
4032	CAA7	4033	CAC5	4034	CAF0	4035	CAFB	4036	CB22
4037	CB39	4038	CB50	4039	CB74	4040	CB98	4041	CBBD
4042	CBEF	4043	CC14	4044	CC38	4045	CC59	4046	CC7A
4047	CCA0	4048	CCC3	4049	CCED	4060	CD0F	4061	CD31
4062	CD80	89	CD9A	99	CDBC	4063	CDCA	4064	CDE6
4065	CDFE	4066	CE17	4067	CE36	4068	CE46	4070	CE65
4071	CE84	4072	CEB0	4073	CED9	4074	CF23	4075	CF6A
4076	CF90	4077	CFC5	4078	D00C	4079	D034	4080	D060
4081	D089	4082	D0A0	4083	D0BF	4084	D108	4085	D118
4086	D136	4087	D14F	4088	D17D	4089	D1A8	4090	D1CF
193	D1FA	9301	D200	9302	D226	9305	D261	9002	D283
9003	D2B5	9004	D2D7	9039	D307	9005	D33C	9007	D38F
9008	D387	9009	D3D9	9010	D409	9020	D450	9041	D496
9045	D4B0	9047	D4E5	2000	D4FF	2001	D526	3002	D557
4050	D566	4051	D590	4052	D5D9	2330	D604	2005	D65?
2009	D669	2010	D699	2020	D6D4	2211	D706	3023	D70F
2241	D724	2025	D794	2026	D7DB	2028	D7FE	2029	D849
2300	D865	3033	D883	2037	D8A0	2044	D8C7	2049	D8FA
2050	D929	2051	D95D	2054	D991	2056	D9CA	2058	DA01

SUBPROGRAMS CALLED

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
IBCOM#	B1D4	UNITX	B1D8	TAPEX	B1DC	DISKX	B1E0	ERRIN	B1E4
[illegible]	B1E8	TOD	B1EC	SWITCH	B1F0	FILE	B1F4	FILEID	B1F8
EXIT	B1FC	IGNSCH	B200	JOHNSN	B204	ERRPRT	B208	MINT	B20C
AMAX1	B210	MAXO	B214						

FORMAT STATEMENT MAP

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
4004	C3EC	4005	C41E	4006	C43B	4007	C491	4008	C4E7
4009	C545	4010	C585	4011	C5CB	4012	C5FF	4013	C673
1	C6C8	2	C6D2	5	C6E3	211	C700	24	C706
33	C718	74	C735	85	C744	4015	C74A	4016	C774
4017	C790	4018	C7C4	4019	C812	4020	C861	4021	C893
4022	C8AB	4023	C8D3	4024	C917	4025	C94F	4026	C998
4027	C9E2	4028	C9FA	4029	CA2F	4030	CA5E	4031	CA8F
4032	CAA7	4033	CAC5	4034	CAF0	4035	CAFB	4036	CB22
4037	CB39	4038	CB5D	4039	CB74	4040	CB98	4041	CBBD
4042	CBFF	4043	CC14	4044	CC3B	4045	CC59	4046	CC7A
4047	CCA0	4048	CCC3	4049	CCED	4060	CD0F	4061	CD31
4062	CD50	89	CD9A	99	CDBC	4063	CDCA	4064	CDE6
4065	CDFE	4066	CE17	4067	CE36	4068	CE46	4070	CE65
4071	CE84	4072	CEB0	4073	CED9	4074	CF23	4075	CF6A
4076	CF90	4077	CFC5	4078	D00C	4079	D034	4080	D060
4081	D089	4082	D0AD	4083	D0BF	4084	D108	4085	D118
4086	D136	4087	D14F	4088	D17D	4089	D1A8	4090	D1CF
193	D1FA	9301	D200	9302	D226	9305	D261	9002	D283
9003	D285	9004	D2D7	9039	D307	9005	D33C	9007	D38C
9008	D387	9009	D3D9	9010	D409	9020	D450	9041	D406
9045	D480	9047	D4E5	2000	D4FF	2001	D526	3002	D557
4050	D565	4051	D590	4052	D5D9	2330	D604	3005	D65?
2009	D669	2010	D699	2020	D6D4	3211	D706	3023	D72F
2241	D724	2025	D79A	2026	D7DB	2028	D7FE	2029	D849
2300	D865	3033	D883	2037	D8A0	2044	D8C7	2040	D8EA
2050	D929	2051	D95D	2054	D991	2056	D9CA	2058	DA01

FORTRAN IV G LEVEL 0, MOD 0 MAIN DATE = 67256 21/30/55 PAGE 0067

2066	DA35	3074	DA63	2750	DA75	2751	DAA2	2762	DACC
2770	DAF8	2077	DB20	2780	DB4D	2078	DB76	2792	DBA4
2079	DBCE	2800	DBED	2081	DC28	3085	DC63	3099	DC6C
2093	DC9F	2940	DCC3	2910	DCFF	2911	DD22	2071	DD40
2972	DD9E	2978	DE01	3099	DE18	2104	DE2C	2109	DE53
2125	DE86	2126	DEA7	2127	DED8	2129	DEDB	2135	DF3D
2154	DF60	3193	DF8C						

TOTAL MEMORY REQUIREMENTS 01CAAA BYTES

```
0001           SUBROUTINE UNITX(TDEV,TMPXMA,TSEL1A,TSEL2A,TSEL3A,TSEL4A,TSEL5A,  S0100001
              1TSEL6A,TCPU,TSETUP,TNOVLP,DEVSPD,DEVRCD,DEVSTP,VOLUME,MULT,FACTOR,S0100002
              2VARINS,AVINST,AVSEL,AVMPX,NOVLP,CUU,D1,D2,D3,D4,D5,D6,X)         S0100003
         C ************************************************************************
         C
         C COMPUTE DEVICE, CHANNEL, CPU, AND SET UP TIMES FOR UNIT RECORD DEVICES
         C
         C ************************************************************************
0002           REAL MULT,NOVLP                                                   S0100005
         C
         C DETERMINE CHANNEL AND INTEFERENCE FOR UNIT
         C
0003           CALL CHNCUU (TMPXMA,TSEL1A,TSEL2A,TSEL3A,TSEL4A,TSEL5A,TSEL6A,    S0100010
              1AVSEL,AVMPX,D1,D2,D3,D4,D5,D6,CUU,1,TCHNA,AVCHN,ICHN)            S0100020
0004           TSETUP=TSETUP+(DEVSTP*60000000)                                   S0100030
         C
         C DOES DEVICE OPERATE AT ITS RATED SPEED
         C
0005           IF(FACTOR-0.)1,1,2                                                S0100040
         C
         C YES.  COMPUTE DEVICE TIME
         C
0006         1 TDEV=(VOLUME*60000000)/DEVSPD                                     S0100050
0007           GO TO 3                                                           S0100060
         C
         C NO.  MULTIPLY DEVICE SPEED BY FACTOR
         C
0008         2 TDEV=(VOLUME*60000000)/(DEVSPD*FACTOR)                            S0100070
0009         3 TCHNA=TCHNA+TDEV                                                  S0100080
0010           TCPU=TCPU+(VOLUME*DEVRCD*AVCHN)                                   S0100090
         C
         C IS THE NUMBER OF INSTRUCTIONS IN THE PROGRAM A FUNCTION OF THIS FILE'S VOLUME
         C
0011           IF(MULT-X)5,4,5                                                   S0100100
         C
         C YES.  MULTIPLY VOLUME BY VARIABLE NUMBER OF INSTRUCTIONS PER RECORD
         C
0012         4 TCPU=TCPU+((VOLUME*VARINS*AVINST)*(1.-NOVLP))                     S0100110
0013           TNOVLP=TNOVLP+(VOLUME*VARINS*AVINST*NOVLP)                        S0100120
         C
         C STORE NEW CHANNEL TIME
         C
0014         5 CALL CHNCUU (TMPXMA,TSEL1A,TSEL2A,TSEL3A,TSEL4A,TSEL5A,TSEL6A,    S0100125
              1AVSEL,AVMPX,D1,D2,D3,D4,D5,D6,CUU,2,TCHNA,AVCHN,ICHN)            S0100130
0015           RETURN                                                            S0100135
0016           END                                                               S0100140
```

SCALAR MAP

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
TMPXMA	B8	TSEL1A	BC	TSEL2A	C0	TSEL3A	C4	TSEL4A	C8
TSEL5A	CC	TSEL6A	D0	AVSEL	D4	AVMPX	D8	D1	DC
D2	E0	D3	E4	D4	E8	D5	EC	D6	F0
CUU	F4	TCHNA	F8	AVCHN	FC	ICHN	100	TSETUP	104
DEVSTP	108	FACTOR	10C	TDEV	110	VOLUME	114	DEVSPD	118
TCPU	11C	DEVRCD	120	MULT	124	X	128	VARINS	12C
AVINST	130	NOVLP	134	TNOVLP	138				

SUBPROGRAMS CALLED

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
CHNCUU	13C								

TOTAL MEMORY REQUIREMENTS 00056C BYTES

```
0001         SUBROUTINE DISKX(TDEV,TMPXBA,TMPXBW,TSEL1A,TSEL1W,TSEL2A,TSEL2W,  S0200001
            1TSEL3A,TSEL3W,TSEL4A,TSEL4W,TSEL5A,TSEL5W,TSEL6A,TSEL6W,TCPU,     S0200002
            2TSETUP,TNOVLP,DEVSPD,DEVSTP,VOLUME,RCDSIZ,BLKFCT,MULT,LTNCY,SEEK, S0200003
            3VARINS,AVINST,AVSEL,AVMPX,NOVLP,CUU,D1,D2,D3,D4,D5,D6,X)          S0200004
      C ***********************************************************************
      C
      C COMPUTE DEVICE, CHANNEL, CPU, AND SET UP TIMES FOR DISK DEVICES
      C
      C ***********************************************************************
0002         REAL MULT,LTNCY,NOVLP                                             S0200005
      C
      C DETERMINE CHANNEL AND INTEFERENCE FOR UNIT
      C
0003         CALL CHNCUU (TMPXBA,TSEL1A,TSEL2A,TSEL3A,TSEL4A,TSEL5A,TSEL6A,    S0200010
            1AVSEL,AVMPX,D1,D2,D3,D4,D5,D6,CUU,1,TCHNA,AVCHN,ICHN)             S0200015
0004         CALL CHNCUU (TMPXBW,TSEL1W,TSEL2W,TSEL3W,TSEL4W,TSEL5W,TSEL6W,    S0200020
            1AVSEL,AVMPX,D1,D2,D3,D4,D5,D6,CUU,1,TCHNW,AVCHN,ICHN)             S0200025
0005         TSETUP=TSETUP+(DEVSTP*60000000)                                   S0200030
0006         TCHNAX=(VOLUME*1000000*RCDSIZ)/DEVSPD                             S0200040
0007         TCHNWX=(VOLUME*1000*(LTNCY+SEEK))/BLKFCT                          S0200050
0008         TDEV=TDEV+TCHNAX+TCHNWX                                           S0200060
0009         IF(TCHNAX-TCHNW)1,2,2                                             S0200070
0010       1 TCHNW=TCHNW-TCHNAX                                                S0200080
0011         GO TO 3                                                           S0200090
0012       2 TCHNW=0.                                                          S0200100
0013       3 IF(TCHNA-TCHNWX)4,5,5                                             S0200110
0014       4 TCHNW=TCHNW+TCHNWX-TCHNA                                          S0200120
0015       5 TCHNA=TCHNA+TCHNAX                                                S0200130
0016         TCPU=TCPU+(VOLUME*RCDSIZ*AVCHN)                                   S0200140
      C
      C IS THE NUMBER OF INSTRUCTIONS IN THE PROGRAM A FUNCTION OF THIS FILE'S VOLUME
      C
0017         IF(MULT-X)7,6,7                                                   S0200150
      C
      C YES.  MULTIPLY VOLUME BY VARIABLE NUMBER OF INSTRUCTIONS PER RECORD
      C
0018       6 TCPU=TCPU+((VOLUME*VARINS*AVINST)*(1.-NOVLP))                     S0200160
0019         TNOVLP=TNOVLP+(VOLUME*VARINS*AVINST*NOVLP)                        S0200170
      C
      C STORE NEW CHANNEL TIME
      C
0020       7 CALL CHNCUU (TMPXBA,TSEL1A,TSEL2A,TSEL3A,TSEL4A,TSEL5A,TSEL6A,    S0200173
            1AVSEL,AVMPX,D1,D2,D3,D4,D5,D6,CUU,2,TCHNA,AVCHN,ICHN)             S0200176
0021         CALL CHNCUU (TMPXBW,TSEL1W,TSEL2W,TSEL3W,TSEL4W,TSEL5W,TSEL6W,    S0200180
            1AVSEL,AVMPX,D1,D2,D3,D4,D5,D6,CUU,2,TCHNW,AVCHN,ICHN)             S0200183
0022         RETURN                                                            S0200186
0023         END                                                               S0200190
```

SCALAR MAP

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
TMPXBA	C8	TSEL1A	CC	TSEL2A	D0	TSEL3A	D4	TSEL4A	D8
TSEL5A	DC	TSEL6A	E0	AVSEL	E4	AVMPX	E8	D1	EC
D2	F0	D3	F4	D4	F8	D5	FC	D6	100
CUU	104	TCHNA	108	AVCHN	10C	ICHN	110	TMPXBW	114
TSEL1W	118	TSEL2W	11C	TSEL3W	120	TSEL4W	124	TSEL5W	128
TSEL6W	12C	TCHNW	130	TSETUP	134	DEVSTP	138	TCHNAX	13C
VOLUME	140	RCDSIZ	144	DEVSPD	148	TCHNWX	14C	LTNCY	150
SEEK	154	BLKFCT	158	TDEV	15C	TCPU	160	MULT	164
X	168	VARINS	16C	AVINST	170	NOVLP	174	TNOVLP	178

SUBPROGRAMS CALLED

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
CHNCUU	17C								

TOTAL MEMORY REQUIREMENTS 00076A BYTES

```
0001         SUBROUTINE TAPEX(TDEV,TMPXBA,TSEL1A,TSEL2A,TSEL3A,TSEL4A,TSEL5A,  S0300001
            1TSEL6A,TCPU,TSETUP,TWNDUP,TNOVLP,DEVSPD,DEVSTP,VOLUME,RCDSIZ,     S0300002
            2BLKFCT,MULT,STRSTP,REWIND,DEVDEN,VARINS,AVINST,AVSEL,AVMPX,NOVLP, S0300003
            3CUU,D1,D2,D3,D4,D5,D6,X)                                          S0300004
      C *************************************************************************
      C
      C COMPUTE DEVICE, CHANNEL, CPU, SET UP, AND REWIND TIMES FOR TAPE DEVICES
      C
      C *************************************************************************
0002         REAL MULT,NOVLP                                                   S0300005
      C
      C DETERMINE CHANNEL AND INTEFERENCE FOR UNIT
      C
0003         CALL CHNCUU (TMPXBA,TSEL1A,TSEL2A,TSEL3A,TSEL4A,TSEL5A,TSEL6A,    S0300010
            1AVSEL,AVMPX,D1,D2,D3,D4,D5,D6,CUU,1,TCHNA,AVCHN,ICHN)             S0200020
0004         TSETUP=TSETUP+(DEVSTP*60000000)                                   S0300040
0005         TDEV=((STRSTP*VOLUME*100)/BLKFCT)+((VOLUME*1000000*RCDSIZ)/DEVSPD)S0300050
0006         TCHNA=TCHNA+TDEV                                                  S0300070
0007         TCPU=TCPU+(VOLUME*RCDSIZ*AVCHN)                                   S0300080
      C
      C IS THE NUMBER OF INSTRUCTIONS IN THE PROGRAM A FUNCTION OF THIS FILE'S VOLUME
      C
0008         IF(MULT-X)2,1,2                                                   S0300090
      C
      C YES.  MULTIPLY VOLUME BY VARIABLE NUMBER OF INSTRUCTIONS PER RECORD
      C
0009       1 TCPU=TCPU+((VOLUME*VARINS*AVINST)*(1.-NOVLP))                     S0300100
0010         TNOVLP=TNOVLP+(VOLUME*VARINS*AVINST*NOVLP)                        S0300110
0011       2 TWND=((((VOLUME*RCDSIZ)/DEVDEN)+(.5*(VOLUME/BLKFCT)))*6000000*    S0300120
            1REWIND)/(2400*12)                                                 S0300130
0012         IF(TWNDUP-TWND)3,4,4                                              S0300140
0013       3 TWNDUP=TWND                                                       S0300150
      C
      C STORE NEW CHANNEL TIME
      C
0014       4 CALL CHNCUU (TMPXBA,TSEL1A,TSEL2A,TSEL3A,TSEL4A,TSEL5A,TSEL6A,    S0300155
            1AVSEL,AVMPX,D1,D2,D3,D4,D5,D6,CUU,2,TCHNA,AVCHN,ICHN)             S0300160
0015         RETURN                                                            S0300165
0016         END                                                               S0300170
```

SCALAR MAP

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
TMPXBA	B0	TSEL1A	B4	TSEL2A	B8	TSEL3A	BC	TSEL4A	C0
TSEL5A	C4	TSEL6A	C8	AVSEL	CC	AVMPX	D0	D1	D4
D2	D8	D3	DC	D4	E0	D5	E4	D6	E8
CUU	EC	TCHNA	F0	AVCHN	F4	ICHN	F8	TSETUP	FC
DEVSTP	100	TDEV	104	STRSTP	108	VOLUME	10C	BLKFCT	110
RCDSIZ	114	DEVSPD	118	TCPU	11C	MULT	120	X	124
VARINS	128	AVINST	12C	NOVLP	130	TNOVLP	134	TWND	138
DEVDEN	13C	REWIND	140	TWNDUP	144				

SUBPROGRAMS CALLED

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
CHNCUU	148								

TOTAL MEMORY REQUIREMENTS 00062E BYTES

```
0001            SUBROUTINE CHNCUU(C0,C1,C2,C3,C4,C5,C6,AVSEL,AVMPX,D1,D2,D3,D4,D5,S0400010
               1D6,CUU,I,TCHN,AVCHN,ICHN)                                        S0400020
          C ***********************************************************************
          C
          C DETERMINE CHANNEL AND INTEFERENCE FOR UNIT -- OR STORE NEW CHANNEL TIME
          C
          C ***********************************************************************
0002            GO TO (20,29),I                                                  S0400030
0003         20 IF(ABS(CUU)-ABS(D1))10,11,21                                     S0400040
0004         21 IF(ABS(CUU)-ABS(D2))11,12,22                                     S0400050
0005         22 IF(ABS(CUU)-ABS(D3))12,13,23                                     S0400060
0006         23 IF(ABS(CUU)-ABS(D4))13,14,24                                     S0400070
0007         24 IF(ABS(CUU)-ABS(D5))14,15,25                                     S0400080
0008         25 IF(ABS(CUU)-ABS(D6))15,16,16                                     S0400090
0009         10 TCHN=C0                                                          S0400100
0010            ICHN=7                                                           S0400110
0011            AVCHN=AVMPX                                                      S0400120
0012            RETURN                                                           S0400130
0013         11 TCHN=C1                                                          S0400140
0014            ICHN=1                                                           S0400150
0015            GO TO 27                                                         S0400160
0016         12 TCHN=C2                                                          S0400170
0017            ICHN=2                                                           S0400180
0018            GO TO 27                                                         S0400190
0019         13 TCHN=C3                                                          S0400200
0020            ICHN=3                                                           S0400210
0021            GO TO 27                                                         S0400220
0022         14 TCHN=C4                                                          S0400230
0023            ICHN=4                                                           S0400240
0024            GO TO 27                                                         S0400250
0025         15 TCHN=C5                                                          S0400260
0026            ICHN=5                                                           S0400270
0027            GO TO 27                                                         S0400280
0028         16 TCHN=C6                                                          S0400290
0029            ICHN=6                                                           S0400300
0030         27 AVCHN=AVSEL                                                      S0400310
0031            RETURN                                                           S0400320
0032         29 GO TO (31,32,33,34,35,36,30),ICHN                                S0400330
0033         30 C0=TCHN                                                          S0400340
0034            RETURN                                                           S0400350
0035         31 C1=TCHN                                                          S0400360
0036            RETURN                                                           S0400370
0037         32 C2=TCHN                                                          S0400380
0038            RETURN                                                           S0400390
0039         33 C3=TCHN                                                          S0400400
0040            RETURN                                                           S0400410
0041         34 C4=TCHN                                                          S0400420
0042            RETURN                                                           S0400430
```

```
0043          35 C5=TCHN                                                  S0400440
0044             RETURN                                                   S0400450
0045          36 C6=TCHN                                                  S0400460
0046             RETURN                                                   S0400470
0047             END                                                      S0400480
```

SCALAR MAP

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
I	140	CUU	144	D1	148	D2	14C	D3	150
D4	154	D5	158	D6	15C	TCHN	160	C0	164
ICHN	168	AVCHN	16C	AVMPX	170	C1	174	C2	178
C3	17C	C4	180	C5	184	C6	188	AVSEL	18C

TOTAL MEMORY REQUIREMENTS 0005FC BYTES

```
0001          SUBROUTINE FRCTN(TOTAL,TCPU,TSEL1A,TSEL1W,TSEL2A,TSEL2W,TSEL3A,   S0500010
             1TSEL3W,TSEL4A,TSEL4W,TSEL5A,TSEL5W,TSEL6A,TSEL6W,TMPXMA,TMPXMW,   S0500020
             2TMPXBA,TMPXBW,TNOVLP,CPU,FRACM,FRACB,FRACC)                       S0500030
        C ***********************************************************************
        C
        C SEPARATE RESOURCE USES INTO MULTIPLEXOR MODE, BURST MODE, AND CPU PORTIONS
        C
        C ***********************************************************************
0002          INTEGER CPU                                                       S0500040
0003          DIMENSION FRACM(12,2),FRACB(12,2),FRACC(12,2)                     S0500050
        C
        C DETERMINE MAXIMUM SELECTOR CHANNEL USAGE
        C
0004          TOTAL=AMAX1(TSEL1A+TSEL1W,TSEL2A+TSEL2W,TSEL3A+TSEL3W,            S0500060
             1TSEL4A+TSEL4W,TSEL5A+TSEL5W,TSEL6A+TSEL6W)                        S0500070
        C
        C ARE THERE ANY MULTIPLEXOR CHANNEL OPERATIONS FOR THIS PROGRAM
        C
0005          IF(TMPXMA+TMPXBA-0.)7,7,10                                        S0500072
        C
        C NO.  SET MULTIPLEXOR MODE FRACTION TO ONE, AND SET BURST MODE FRACTION TO ZERO
        C
0006        7 RATIO1=1.                                                         S0500074
0007          RATIO2=0.                                                         S0500076
0008          GO TO 15                                                          S0500078
        C
        C YES.  COMPUTE MULTIPLEXOR MODE AND BURST MODE FRACTIONS
        C
0009       10 RATIO1=TMPXMA/(TMPXMA+TMPXBA)                                     S0500079
0010          RATIO2=TMPXBA/(TMPXMA+TMPXBA)                                     S0500080
        C
        C ARE THERE ANY BURST MODE OPERATIONS FOR THIS PROGRAM
        C
0011          IF(RATIO2-0.)15,15,20                                             S0500082
        C
        C NO.  SET NON-OVERLAPPED BURST MODE AND NON-OVERLAPPED CPU FRACTIONS TO ZERO
        C
0012       15 RATIO3=0.                                                         S0500084
0013          RATIO4=0.                                                         S0500086
0014          GO TO 25                                                          S0500088
        C
        C YES.  COMPUTE NON-OVERLAPPED BURST MODE AND NON-OVERLAPPED CPU FRACTIONS
        C
0015       20 RATIO3=TMPXBA/(TMPXBA+(TCPU*RATIO2))                              S0500089
0016          RATIO4=(TCPU*RATIO2)/(TMPXBA+(TCPU*RATIO2))                       S0500090
        C
        C CAN BURST MODE OPERATIONS BE OVERLAPPED WITH INSTRUCTION EXECUTION
        C
```

```
0017          25 IF(CPU-2040)1,1,2                                                    S0500100
            C
            C NO.  COMPUTE TOTAL PROGRAM TIME
            C
0018            1 TOTAL=AMAX1(TOTAL,AMAX1(TMPXMA,TMPXBW,TCPU*RATIO1)+                 S0500110
               1AMAX1(TMPXBA,TMPXMW*RATIO3)+AMAX1(TCPU*RATIO2,TMPXMW*RATIO4))+         S0500115
               2TNOVLP                                                                S0500120
0019              GO TO 3                                                             S0500130
            C
            C YES.  COMPUTE TOTAL PROGRAM TIME
            C
0020            2 TOTAL=AMAX1(TOTAL,AMAX1(TMPXMA,TMPXBW,TCPU*RATIO1)+                 S0500140
               1AMAX1(TMPXBA,TMPXMW,TCPU*RATIO2))+TNOVLP                              S0500150
            C
            C SET UP POINTERS FOR RESOURCES
            C
0021            3 DO 4 I=1,12                                                         S0500160
0022              FRACM(I,1)=I-1                                                      S0500170
0023              FRACB(I,1)=I-1                                                      S0500180
0024            4 FRACC(I,1)=I-1                                                      S0500190
            C
            C ARE THERE ANY MULTIPLEX MODE OPERATIONS FOR THIS PROGRAM
            C
0025              IF(RATIO1-0.)40,40,45                                               S0500192
            C
            C NO.  SET MULTIPLEX MODE PORTION TO ZERO
            C
0026           40 DO 41 I=1,12                                                        S0500194
0027           41 FRACM(I,2)=0.                                                       S0500196
0028              GO TO 49                                                            S0500198
            C
            C YES.  COMPUTE FRACTIONS FOR MULTIPLEX MODE PORTION
            C
0029           45 FRACM(1,2)=TMPXMA/TOTAL                                             S0500200
0030              FRACM(8,2)=0.                                                       S0500210
0031              FRACM(9,2)=(TCPU*RATIO1)/TOTAL                                      S0500220
0032              CALL FRCTBL (FRACM,TOTAL/RATIO1,TSEL1A,TSEL2A,TSEL3A,TSEL4A,        S0500230
                 1TSEL5A,TSEL6A,TNOVLP,RATIO1)                                        S0500240
            C
            C ARE THERE ANY BURST MODE OPERATIONS FOR THIS PROGRAM
            C
0033           49 IF(RATIO2-0.)50,50,55                                               S0500242
            C
            C NO.  SET BURST MODE PORTION TO ZERO
            C
0034           50 DO 51 I=1,12                                                        S0500244
0035           51 FRACB(I,2)=0.                                                       S0500246
            C
```

```
             C CAN BURST MODE OPERATIONS BE OVERLAPPED WITH INSTRUCTION EXECUTION
             C                                                                    S0500248
 0036             IF(CPU-2040)60,60,59
             C
             C YES.  RETURN TO MAIN PROGRAM
             C                                                                    S0500249
 0037          59 RETURN
             C
             C BURST MODE OPERATIONS EXIST.  COMPUTE FRACTIONS FOR BURST MODE PORTION
             C                                                                    S0500250
 0038          55 FRACB(1,2)=0.                                                   S0500260
 0039             FRACB(8,2)=IMPXBA/TOTAL
             C
             C CAN BURST MODE OPERATIONS BE OVERLAPPED WITH INSTRUCTION EXECUTION
             C                                                                    S0500270
 0040             IF(CPU-2040)5,5,6
             C
             C NO.  SET CPU FRACTION FOR BURST MODE PORTION EQUAL TO ZERO
             C                                                                    S0500280
 0041           5 FRACB(9,2)=0.                                                   S0500290
 0042             CALL FRCTBL (FRACB,TOTAL/(RATIO3*RATIO2),TSEL1A,TSEL2A,TSEL3A,  S0500300
                 1TSEL4A,TSEL5A,TSEL6A,TNOVLP,RATIO3*RATIO2)
             C
             C ARE THERE ANY CPU OPERATIONS
             C                                                                    S0500302
 0043             IF(RATIO4-0.)60,60,65
             C
             C NO.  SET CPU PORTION TO ZERO
             C                                                                    S0500304
 0044          60 DO 61 I=1,12                                                    S0500306
 0045          61 FRACC(I,2)=0.                                                   S0500308
 0046             RETURN
             C
             C YES.  COMPUTE FRACTIONS FOR CPU PORTION
             C                                                                    S0500310
 0047          65 FRACC(1,2)=0.                                                   S0500320
 0048             FRACC(8,2)=0.                                                   S0500330
 0049             FRACC(9,2)=(TCPU*RATIO2)/TOTAL                                  S0500340
 0050             CALL FRCTBL (FRACC,TOTAL/(RATIO4*RATIO2),TSEL1A,TSEL2A,TSEL3A,  S0500350
                 1TSEL4A,TSEL5A,TSEL6A,TNOVLP,RATIO4*RATIO2)                      S0500360
 0051             RETURN
             C
             C BURST MODE OPERATIONS AND INSTRUCTION EXECUTION CAN BE OVERLAPPED
             C                                                                    S0500370
 0052           6 FRACB(9,2)=(TCPU*RATIO2)/TOTAL                                  S0500380
 0053             CALL FRCTBL (FRACB,TOTAL/RATIO2,TSEL1A,TSEL2A,TSEL3A,TSEL4A,    S0500390
                 1TSEL5A,TSEL6A,TNOVLP,RATIO2)                                    S0500400
 0054             RETURN
```

```
0055             END                                              S0500410
```

SCALAR MAP

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
TOTAL	130	TSEL1A	134	TSEL1W	138	TSEL2A	13C	TSEL2W	140
TSEL3A	144	TSEL3W	148	TSEL4A	14C	TSEL4W	150	TSEL5A	154
TSEL5W	158	TSEL6A	15C	TSEL6W	160	TMPXMA	164	TMPXBA	168
RATIO1	16C	RATIO2	170	RATIO3	174	RATIO4	178	TCPU	17C
CPU	180	TMPXBW	184	TMPXMW	188	TNOVLP	18C	I	190

ARRAY MAP

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
FRACM	194	FRACB	198	FRACC	19C				

SUBPROGRAMS CALLED

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
FRCTBL	1A0	AMAX1	1A4						

TOTAL MEMORY REQUIREMENTS 000996 BYTES

```
0001          SUBROUTINE FRCTBL(FRAC,DENOM,TSEL1A,TSEL2A,TSEL3A,TSEL4A,TSEL5A,  S0600010
             1TSEL6A,TNOVLP,RATIO)                                              S0600020
        C ********************************************************************
        C
        C COMPLETE COMPUTATION OF FRACTIONS FOR THIS PORTION AND THEN SORT FRACTIONS
        C
        C ********************************************************************
0002          DIMENSION FRAC(12,2)                                              S0600030
        C
        C COMPUTE SELECTOR CHANNEL FRACTIONS
        C
0003          FRAC(2,2)=TSEL1A/DENOM                                            S0600040
0004          FRAC(3,2)=TSEL2A/DENOM                                            S0600050
0005          FRAC(4,2)=TSEL3A/DENOM                                            S0600060
0006          FRAC(5,2)=TSEL4A/DENOM                                            S0600070
0007          FRAC(6,2)=TSEL5A/DENOM                                            S0600080
0008          FRAC(7,2)=TSEL6A/DENOM                                            S0600090
        C
        C COMPUTE NON-OVERLAPPED CPU FRACTION
        C
0009          FRAC(11,2)=TNOVLP/DENOM                                           S0600100
        C
        C COMPUTE WAIT FRACTION
        C
0010          FRAC(12,2)=AMAX1(0.,RATIO-FRAC(11,2)-AMAX1(FRAC(1,2),FRAC(2,2),   S0600110
             1FRAC(3,2),FRAC(4,2),FRAC(5,2),FRAC(6,2),FRAC(7,2),FRAC(8,2),      S0600120
             2FRAC(9,2)))                                                       S0600130
        C
        C SORT FRACTIONS
        C
0011          DO 3 I=1,8                                                        S0600160
0012          K=I+1                                                             S0600170
0013          DO 3 J=K,9                                                        S0600180
0014          IF(FRAC(J,2)-FRAC(I,2))3,3,4                                      S0600210
0015        4 FRAC(10,1)=FRAC(I,1)                                              S0600290
0016          FRAC(10,2)=FRAC(I,2)                                              S0600300
0017          FRAC(I,1)=FRAC(J,1)                                               S0600310
0018          FRAC(I,2)=FRAC(J,2)                                               S0600320
0019          FRAC(J,1)=FRAC(10,1)                                              S0600330
0020          FRAC(J,2)=FRAC(10,2)                                              S0600340
0021        3 CONTINUE                                                          S0600350
        C
        C MARK END OF TABLE FOR OVERLAPPED RESOURCE USES
        C
0022          FRAC(10,1)=9.                                                     S0600360
0023          FRAC(10,2)=0.                                                     S0600370
0024          RETURN                                                            S0600380
0025          END                                                               S0600390
```

SCALAR MAP

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
TSEL1A	A8	DENOM	AC	TSEL2A	B0	TSEL3A	B4	TSEL4A	B8
TSEL5A	BC	TSEL6A	C0	TNOVLP	C4	RATIO	C8	I	CC
K	D0	J	D4						

ARRAY MAP

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
FRAC	D8								

SUBPROGRAMS CALLED

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
AMAX1	DC								

TOTAL MEMORY REQUIREMENTS 0003E8 BYTES

```
0001          SUBROUTINE TIME2(FRAC1,FRAC2,FRAC12,CPU,TIMEB)                   S0700010
        C ***********************************************************************
        C
        C CALCULATE MULTIPROGRAMMING TIME FOR SECOND PRIORITY PROGRAM
        C
        C ***********************************************************************
0002          INTEGER CPU                                                      S0700020
0003          DIMENSION FRAC1(12,2),FRAC2(12,2),FRAC12(11,2)                   S0700030
0004          DO 2 J=1,11                                                      S0700032
0005          DO 2 K=1,2                                                       S0700034
0006        2 FRAC12(J,K)=0.                                                   S0700036
0007          I=1                                                              S0700040
        C
        C FIND THE NUMBER OF RESOURCES USED BY THE FIRST PROGRAM
        C
0008          DO 1 M=1,10                                                      S0700050
0009          IF(FRAC1(M,2)-0.)100,100,1                                       S0700060
0010        1 CONTINUE                                                         S0700070
0011      100 M=M-1                                                            S0700080
        C
        C IS THIS NUMBER ZERO
        C
0012          IF(M-0)800,800,110                                               S0700085
        C
        C NO.  FIND THE NUMBER OF RESOURCES USED BY THE SECOND PROGRAM
        C
0013      110 DO 101 N=1,10                                                    S0700090
0014          IF(FRAC2(N,2)-0.)200,200,101                                     S0700100
0015      101 CONTINUE                                                         S0700110
0016      200 N=N-1                                                            S0700120
        C
        C IS THIS NUMBER ZERO
        C
0017          IF(N-0)800,800,210                                               S0700125
        C
        C NO.  DETERMINE IF ALL RESOURCE USES BY BOTH PROGRAMS CAN BE OVERLAPPED
        C
0018      210 DO 201 J=1,N                                                     S0700130
        C
        C IS PROGRAM TWO'S RESOURCE THE MULTIPLEXOR CHANNEL IN MULTIPLEX MODE
        C
0019          IF(FRAC2(J,1)-0.)300,300,220                                     S0700140
        C
        C NO.  IS PROGRAM TWO'S RESOURCE THE MULTIPLEXOR CHANNEL IN BURST MODE
        C
0020      220 IF(FRAC2(J,1)-7.)500,400,230                                     S0700150
        C
        C NO.  CAN BURST MODE MPX OPERATIONS BE OVERLAPPED WITH INSTRUCTION EXECUTION
```

```
              C
0021             230 IF(CPU-2040)240,240,500                                      S0700160
              C
              C NO.  IS PROGRAM TWO'S RESOURCE THE CPU
              C
0022             240 IF(FRAC2(J,1)-8.)500,250,500                                 S0700170
              C
              C YES.  IS THERE A RESOURCE CONFLICT BETWEEN PROGRAM TWO AND PROGRAM ONE
              C
0023             250 DO 251 K=1,M                                                 S0700180
0024                 IF(FRAC1(K,1)-7.)251,600,260                                 S0700190
0025             260 IF(FRAC1(K,1)-8.)251,600,251                                 S0700200
0026             251 CONTINUE                                                     S0700210
              C
              C NO CONFLICT.  CHECK NEXT PROGRAM TWO RESOURCE
              C
0027                 GO TO 201                                                    S0700230
              C
              C PROGRAM TWO USES MPX CHANNEL IN MPX MODE.  IS THERE A RESOURCE CONFLICT
              C
0028             300 DO 301 K=1,M                                                 S0700240
0029                 IF(FRAC1(K,1)-7.)301,600,301                                 S0700250
0030             301 CONTINUE                                                     S0700260
              C
              C NO CONFLICT.  CHECK NEXT PROGRAM TWO RESOURCE
              C
0031                 GO TO 201                                                    S0700270
              C
              C PROGRAM TWO USES MPX CHANNEL IN BURST MODE.  IS THERE A RESOURCE CONFLICT
              C
0032             400 DO 401 K=1,M                                                 S0700280
0033                 IF(FRAC1(K,1)-0.)600,600,410                                 S0700290
0034             410 IF(FRAC1(K,1)-7.)401,600,420                                 S0700300
0035             420 IF(CPU-2040)430,430,401                                      S0700310
0036             430 IF(FRAC1(K,1)-8.)401,600,401                                 S0700320
0037             401 CONTINUE                                                     S0700330
              C
              C NO CONFLICT.  CHECK NEXT PROGRAM TWO RESOURCE
              C
0038                 GO TO 201                                                    S0700340
              C
              C ARE BOTH PROGRAMS USING THE CPU OR THE SAME SELECTOR CHANNEL
              C
0039             500 DO 501 K=1,M                                                 S0700350
0040                 IF(FRAC1(K,1)-FRAC2(J,1))501,600,501                         S0700360
0041             501 CONTINUE                                                     S0700370
              C
              C NO.  CHECK NEXT PROGRAM TWO RESOURCE
```

```
                C
0042               201 CONTINUE                                                        S0700380
                C
                C NO RESOURCE CONFLICT.  SAVE NUMBER OF RESOURCES FOR EACH PROGRAM AND
                C INCREMENT PROGRAM TWO'S TIME BY PRODUCT OF TIMES FOR THE TWO SETS OF RESOURCES
                C
0043                  FRAC12(I,1)=M                                                     S0700390
0044                  FRAC12(I,2)=N                                                     S0700400
0045                  TIMEB=TIMEB+((FRAC1(M,2)-FRAC1(M+1,2))*(FRAC2(1,2)-FRAC2(N+1,2))) S0700405
0046                  I=I+1                                                             S0700410
0047                  GO TO 700                                                         S0700420
                C
                C RESOURCE CONFLICT.  REMOVE A RESOURCE FROM PROGRAM TWO
                C
0048               600 N=N-1                                                           S0700430
                C
                C IS THE NUMBER OF RESOURCES EQUAL TO ZERO
                C
0049                  IF(N-0)700,700,210                                                S0700440
                C
                C YES.  REMOVE A RESOURCE FROM PROGRAM ONE
                C
0050               700 M=M-1                                                           S0700450
                C
                C IS THE NUMBER OF RESOURCES EQUAL TO ZERO
                C
0051                  IF(M-0)800,800,110                                                S0700460
                C
                C YES.  MARK END OF TABLE
                C
0052               800 FRAC12(I,1)=0.                                                  S0700470
0053                  FRAC12(I,2)=0.                                                    S0700480
                C
                C SAVE POINTER TO PROGRAM ONE'S OVERLAPPED CPU TIME
                C
0054                  DO 900 L=1,9                                                      S0700490
0055                  IF(FRAC1(L,1)-8.)900,901,900                                      S0700500
0056               900 CONTINUE                                                        S0700510
0057               901 FRAC12(11,1)=L                                                  S0700520
                C
                C INCREMENT PROGRAM TWO'S TIME BY ITS NON-CONFLICT NON-OVERLAPPED CPU TIME
                C
0058                  TIMEB=TIMEB+((FRAC2(11,2))*(FRAC1(1,2)-FRAC1(L,2)))               S0700525
                C
                C SAVE POINTER TO PROGRAM TWO'S OVERLAPPED CPU TIME
                C
0059                  DO 1000 L=1,9                                                     S0700530
0060                  IF(FRAC2(L,1)-8.)1000,1001,1000                                   S0700540
```

```
0061        1000 CONTINUE                                                              S0700550
0062        1001 FRAC12(11,2)=L                                                        S0700560
           C
           C INCREMENT TIME FOR PGM TWO BY PGM ONE'S NON-CONFLICT NON-OVERLAPPED CPU TIME
           C
0063             TIMEB=TIMEB+((FRAC1(11,2))*(FRAC2(1,2)-FRAC2(L,2)))                    S0700565
           C
           C INCREMENT TIME FOR PROGRAM TWO BY PROGRAM ONE'S WAIT TIME
           C
0064             TIMEB=TIMEB+FRAC1(12,2)                                                S0700566
0065             RETURN                                                                 S0700570
0066             END                                                                    S0700580
```

SCALAR MAP

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
J	1A4	K	1A8	I	1AC	M	1B0	N	1B4
CPU	1B8	TIMEB	1BC	L	1C0				

ARRAY MAP

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
FRAC1	1C4	FRAC2	1C8	FRAC12	1CC				

TOTAL MEMORY REQUIREMENTS 000784 BYTES

```
0001          SUBROUTINE TIME3(FRAC1,FRAC2,FRAC3,FRAC12,CPU,TIMEB,TIMEC)         S0800010
        C ****************************************************************************
        C
        C CALCULATE MULTIPROGRAMMING TIME FOR THIRD PRIORITY PROGRAM
        C
        C ****************************************************************************
0002          INTEGER CPU                                                        S0800020
0003          DIMENSION FRAC1(12,2),FRAC2(12,2),FRAC3(12,2),FRAC12(11,2)         S0800030
0004          DIMENSION FRAC23(11,2)                                             S0800040
0005          TIMED=0.                                                           S0800060
0006          DO 700 I=1,10                                                      S0800070
        C
        C IS THERE A NO CONFLICT PROGRAM ONE-PROGRAM TWO RESOURCE COMBINATION
        C
0007          IF(FRAC12(I,1)-0.)800,800,110                                      S0800080
        C
        C YES.  CALCULATE PROGRAM THREE MULTIPROGRAMMING TIME FOR THAT COMBINATION
        C
0008      110 DO 101 N=1,10                                                      S0800090
        C
        C FIND THE NUMBER OF RESOURCES USED BY THE THIRD PROGRAM
        C
0009          IF(FRAC3(N,2)-0.)200,200,101                                       S0800100
0010      101 CONTINUE                                                           S0800110
0011      200 N=N-1                                                              S0800120
0012          DO 202 L=1,2                                                       S0800130
        C
        C FIND THE NUMBER OF RESOURCES USED BY ONE OF THE OTHER PROGRAMS
        C
0013          M=FRAC12(I,L)                                                      S0800140
        C
        C IS IT PROGRAM ONE OR PROGRAM TWO
        C
0014          GO TO (210,5210),L                                                 S0800150
        C
        C PROGRAM ONE.  DETERMINE IF RESOURCE USES OF PGM 1 AND PGM 3 CAN BE OVERLAPPED
        C
0015      210 DO 201 J=1,N                                                       S0800160
        C
        C IS PROGRAM THREE'S RESOURCE THE MULTIPLEXOR CHANNEL IN MULTIPLEX MODE
        C
0016          IF(FRAC3(J,1)-0.)300,300,220                                       S0800170
        C
        C NO.  IS PROGRAM THREE'S RESOURCE THE MULTIPLEXOR CHANNEL IN BURST MODE
        C
0017      220 IF(FRAC3(J,1)-7.)500,400,230                                       S0800180
        C
        C NO.  CAN BURST MODE MPX OPERATIONS BE OVERLAPPED WITH INSTRUCTION EXECUTION
```

```
                C
0018              230 IF(CPU-2040)240,240,500                                       S0800190
                C
                C NO.  IS PROGRAM THREE'S RESOURCE THE CPU
                C
0019              240 IF(FRAC3(J,1)-8.)500,250,500                                  S0800200
                C
                C YES.  IS THERE A RESOURCE CONFLICT BETWEEN PROGRAM THREE AND PROGRAM ONE
                C
0020              250 DO 251 K=1,M                                                  S0800210
0021                  IF(FRAC1(K,1)-7.)251,600,260                                  S0800220
0022              260 IF(FRAC1(K,1)-8.)251,600,251                                  S0800230
0023              251 CONTINUE                                                      S0800240
                C
                C NO CONFLICT.  CHECK NEXT PROGRAM THREE RESOURCE
                C
0024                  GO TO 201                                                     S0800250
                C
                C PROGRAM THREE USES THE MPX CHANNEL IN MPX MODE.  IS THERE A RESOURCE CONFLICT
                C
0025              300 DO 301 K=1,M                                                  S0800260
0026                  IF(FRAC1(K,1)-7.)301,600,301                                  S0800270
0027              301 CONTINUE                                                      S0800280
                C
                C NO CONFLICT.  CHECK NEXT PROGRAM THREE RESOURCE
                C
0028                  GO TO 201                                                     S0800290
                C
                C PROGRAM THREE USES MPX CHANNEL IN BURST MODE.  IS THERE A RESOURCE CONFLICT
                C
0029              400 DO 401 K=1,M                                                  S0800300
0030                  IF(FRAC1(K,1)-0.)600,600,410                                  S0800310
0031              410 IF(FRAC1(K,1)-7.)401,600,420                                  S0800320
0032              420 IF(CPU-2040)430,430,401                                       S0800330
0033              430 IF(FRAC1(K,1)-8.)401,600,401                                  S0800340
0034              401 CONTINUE                                                      S0800350
                C
                C NO CONFLICT.  CHECK NEXT PROGRAM THREE RESOURCE
                C
0035                  GO TO 201                                                     S0800360
                C
                C ARE BOTH PROGRAMS USING THE CPU OR THE SAME SELECTOR CHANNEL
                C
0036              500 DO 501 K=1,M                                                  S0800370
0037                  IF(FRAC1(K,1)-FRAC3(J,1))501,600,501                          S0800380
0038              501 CONTINUE                                                      S0800390
                C
                C NO.  CHECK NEXT PROGRAM THREE RESOURCE
```

```
                C
 0039             201 CONTINUE                                                  S0800400
                C
                C NO RESOURCE CONFLICT BETWEEN PROGRAMS THREE AND ONE
                C SAVE NUMBER OF RESOURCES FOR PROGRAM THREE
                C
 0040                 NX=N                                                      S0800405
 0041                 GO TO 202                                                 S0800410
                C
                C RESOURCE CONFLICT.  REMOVE A RESOURCE FROM PROGRAM THREE
                C
 0042             600 N=N-1                                                     S0800420
                C
                C IS THE NUMBER OF RESOURCES EQUAL TO ZERO
                C
 0043                 IF(N-0)700,700,210                                        S0800430
                C
                C PROGRAM TWO.  DETERMINE IF RESOURCE USES OF PGM 2 AND PGM 3 CAN BE OVERLAPPED
                C
 0044            5210 DO 5201 J=1,N                                             S0800440
                C
                C IS PROGRAM THREE'S RESOURCE THE MULTIPLEXOR CHANNEL IN MULTIPLEX MODE
                C
 0045                 IF(FRAC3(J,1)-0.)5300,5300,5220                           S0800450
                C
                C NO.  IS PROGRAM THREE'S RESOURCE THE MULTIPLEXOR CHANNEL IN BURST MODE
                C
 0046            5220 IF(FRAC3(J,1)-7.)5500,5400,5230                           S0800460
                C
                C NO.  CAN BURST MODE MPX OPERATIONS BE OVERLAPPED WITH INSTRUCTION EXECUTION
                C
 0047            5230 IF(CPU-2040)5240,5240,5500                                S0800470
                C
                C NO.  IS PROGRAM THREE'S RESOURCE THE CPU
                C
 0048            5240 IF(FRAC3(J,1)-8.)5500,5250,5500                           S0800480
                C
                C YES.  IS THERE A RESOURCE CONFLICT BETWEEN PROGRAM THREE AND PROGRAM TWO
                C
 0049            5250 DO 5251 K=1,M                                             S0800490
 0050                 IF(FRAC2(K,1)-7.)5251,5600,5260                           S0800500
 0051            5260 IF(FRAC2(K,1)-8.)5251,5600,5251                           S0800510
 0052            5251 CONTINUE                                                  S0800520
                C
                C NO CONFLICT.  CHECK NEXT PROGRAM THREE RESOURCE
                C
 0053                 GO TO 5201                                                S0800530
                C
```

```
            C PROGRAM THREE USES THE MPX CHANNEL IN MPX MODE.  IS THERE A RESOURCE CONFLICT
            C
 0054        5300 DO 5301 K=1,M                                                      S0800540
 0055             IF(FRAC2(K,1)-7.)5301,5600,5301                                    S0800550
 0056        5301 CONTINUE                                                           S0800560
            C
            C NO CONFLICT.  CHECK NEXT PROGRAM THREE RESOURCE
            C
 0057             GO TO 5201                                                         S0800570
            C
            C PROGRAM THREE USES MPX CHANNEL IN BURST MODE.  IS THERE A RESOURCE CONFLICT
            C
 0058        5400 DO 5401 K=1,M                                                      S0800580
 0059             IF(FRAC2(K,1)-0.)5600,5600,5410                                    S0800590
 0060        5410 IF(FRAC1(K,1)-7.)5401,5600,5420                                    S0800600
 0061        5420 IF(CPU-2040)5430,5430,5401                                         S0800610
 0062        5430 IF(FRAC2(K,1)-8.)5401,5600,5401                                    S0800620
 0063        5401 CONTINUE                                                           S0800630
            C
            C NO CONFLICT.  CHECK NEXT PROGRAM THREE RESOURCE
            C
 0064             GO TO 5201                                                         S0800640
            C
            C ARE BOTH PROGRAMS USING THE CPU OR THE SAME SELECTOR CHANNEL
            C
 0065        5500 DO 5501 K=1,M                                                      S0800650
 0066             IF(FRAC2(K,1)-FRAC3(J,1))5501,5600,5501                            S0800660
 0067        5501 CONTINUE                                                           S0800670
            C
            C NO.  CHECK NEXT PROGRAM THREE RESOURCE
            C
 0068        5201 CONTINUE                                                           S0800680
 0069             GO TO 202                                                          S0800690
            C
            C RESOURCE CONFLICT.  REMOVE A RESOURCE FROM PROGRAM THREE
            C
 0070        5600 N=N-1                                                              S0800700
            C
            C IS THE NUMBER OF RESOURCES EQUAL TO ZERO
            C
 0071             IF(N-0)5700,5700,5210                                              S0800710
            C
            C YES.  REMOVE A RESOURCE FROM PROGRAM TWO
            C
 0072        5700 M=M-1                                                              S0800712
            C
            C IS THE NUMBER OF RESOURCES EQUAL TO ZERO
            C
```

```
0073          IF(M-0)700,700,5710                                              S0800714
        C
        C NO.  RESET PROGRAM THREE'S RESOURCE COUNT
        C
0074     5710 N=NX                                                             S0800716
0075          GO TO 5210                                                       S0800718
0076      202 CONTINUE                                                         S0800720
        C
        C NO RESOURCE CONFLICT AMONG THE THREE PROGRAMS.  INCREMENT PROGRAM THREE'S TIME
        C BY PRODUCT OF TIMES FOR THE THREE SETS OF RESOURCES
        C
0077          J=FRAC12(I,1)                                                    S0800730
0078          TIMEC=TIMEC+((FRAC1(J,2)-FRAC1(J+1,2))*(FRAC2(1,2)-FRAC2(M+1,2))* S0800740
             1(FRAC3(1,2)-FRAC3(N+1,2)))                                       S0800750
        C
        C ARE THERE ANY MORE NO CONFLICT PROGRAM ONE-PROGRAM TWO RESOURCE COMBINATIONS
        C
0079      700 CONTINUE                                                         S0800760
        C
        C NO. INCREMENT PROGRAM THREE'S TIME BY ITS NON-CONFLICT NON-OVERLAPPED CPU TIME
        C
0080      800 J=FRAC12(11,1)                                                   S0800770
0081          M=FRAC12(11,2)                                                   S0800780
0082          TIMEC=TIMEC+((FRAC1(1,2)-FRAC1(J,2))*(FRAC2(1,2)-FRAC2(M,2))*    S0800790
             1(FRAC3(11,2)))                                                   S0800800
        C
        C INCREMENT PROGRAM THREE'S TIME BY PROGRAM ONE'S WAIT TIME
        C
0083          IF(FRAC1(12,2)-0.)805,805,804                                    S0800810
0084      804 CALL TIME2 (FRAC2,FRAC3,FRAC23,CPU,TIMED)                        S0800820
0085          TIMEC=TIMEC+(TIMED*FRAC1(12,2))                                  S0800830
        C
        C INCREMENT PROGRAM THREE'S TIME BY PROGRAM TWO'S WAIT TIME
        C
0066      805 TIMEC=TIMEC+(TIMEB*FRAC2(12,2))                                  S0800840
0087          RETURN                                                           S0800850
0088          END                                                              S0800860
```

SCALAR MAP

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
TIMED	234	I	238	N	23C	L	240	M	244
J	248	CPU	24C	K	250	NX	254	TIMEC	258
TIMEB	25C								

ARRAY MAP

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
FRAC1	260	FRAC2	264	FRAC3	268	FRAC12	26C	FRAC23	270

SUBPROGRAMS CALLED

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
TIME2	2C8								

TOTAL MEMORY REQUIREMENTS 000A44 BYTES

```
0001            SUBROUTINE FILE(D1,D2,A0,A1,A2,MFILE,FILNAM,FILPNT,MCUU,CUU,       S0900010
               1CUUPNT,DEVCLS,BLANK4,BLANK8,D,ERROR,D3,D4)                        S0900015
          C ************************************************************************
          C
          C READ AND PROCESS JBD AND JFD STATEMENTS
          C
          C ************************************************************************
0002            INTEGER FILPNT(20),CUUPNT(20),ERROR                                S0900017
0003            DIMENSION CUU(20),DEVCLS(10),A(20),C(7)                            S0900018
0004            DOUBLE PRECISION FILNAM(20),BLANK8,B(7)                            S0900019
0005            I1=1                                                               S0900020
0006            J1=MIN0(7,MFILE)                                                   S0900030
          C
          C READ A JBD OR JFD STATEMENT
          C
0007          1 READ (5,2) A3,A4,A5,A6,N2,(B(K),C(K),K=1,7),A7                     S0900040
0008          2 FORMAT(4A1,1X,I1,7(A7,A3),3X,A1)                                   S0900050
0009            IF(A3-A0)1002,3,1002                                               S0900060
0010          3 IF(A4-A1)1003,4,1003                                               S0900070
0011          4 IF(A5-A2)1004,5,1004                                               S0900110
0012          5 IF(A6-D)1005,6,1005                                                S0900120
          C
          C CHECK FOR VALID NUMBER OF FILE DEFINITIONS ON STATEMENT
          C
0013          6 IF(MFILE-I1-N2+1)1006,7,7                                          S0900140
0014          7 IF(7-N2)1007,8,8                                                   S0900150
0015          8 IF(N2-1)1008,85,85                                                 S0900155
0016         85 DO 81 K=I1,J1                                                      S0900156
0017            FILNAM(K)=B(K-I1+1)                                                S0900157
0018         81 A(K)=C(K-I1+1)                                                     S0900158
0019            K1=I1+N2-1                                                         S0900160
0020            DO 135 L1=I1,K1                                                    S0900170
0021            IF(L1-2)100,9,9                                                    S0900180
0022          9 K2=L1-1                                                            S0900190
          C
          C CHECK FOR DUPLICATE FILE NAME
          C
0023            DO 10 L2=1,K2                                                      S0900200
0024            IF(FILNAM(L1)-FILNAM(L2))10,1009,10                                S0900210
0025         10 CONTINUE                                                           S0900220
          C
          C RELATE FILE NAME TO UNIT NUMBER
          C
0026        100 DO 11 L3=1,MCUU                                                    S0900230
0027            IF(A(L1)-CUU(L3))11,12,11                                          S0900240
0028         11 CONTINUE                                                           S0900250
0029            GO TO 1011                                                         S0900260
          C
```

```
              C SAVE POINTER TO UNIT NUMBER
              C
0030             12 FILPNT(L1)=L3                                                 S0900270
0031                IF(L1-2)135,120,120                                           S0900271
0032            120 K3=L1-1                                                       S0900272
0033                DO 13 L4=1,K3                                                 S0900273
              C
              C IS A FILE ALREADY ASSIGNED TO THIS UNIT NUMBER
              C
0034                IF(FILPNT(L1)-FILPNT(L4))13,121,13                            S0900274
0035            121 L5=CUUPNT(L3)                                                 S0900275
              C
              C YES.  IS THE UNIT A DISK DEVICE
              C
0036                IF(DEVCLS(L5)-D)1121,135,1121                                 S0900276
0037             13 CONTINUE                                                      S0900280
              C
              C YES.  ARE THERE ANY MORE FILES ON THIS STATEMENT
              C
0038            135 CONTINUE                                                      S0900285
              C
              C NO.  CHECK FOR ADDITIONAL STATEMENTS OF THIS TYPE
              C
0039            130 IF(A7-BLANK4)15,14,15                                         S0900290
              C
              C NO MORE STATEMENTS OF THIS TYPE.  SET REMAINING PLACES IN FILE TABLE TO BLANK
              C
0040             14 IF(I1+N2-MFILE)140,140,150                                    S0900292
0041            140 J1=I1+N2                                                      S0900294
0042                DO 141 K1=J1,MFILE                                            S0900296
0043            141 FILNAM(K1)=BLANK8                                             S0900298
0044            150 RETURN                                                        S0900300
0045             15 I1=I1+N2                                                      S0900310
0046                IF(I1-MFILE)16,16,150                                         S0900315
0047             16 J1=MIN0(I1+6,MFILE)                                           S0900320
0048                GO TO 1                                                       S0900330
              C ***********************************************************************
              C
              C ERROR MESSAGES
              C
              C ***********************************************************************
0049           2000 FORMAT(18H PROGRAM EXPECTED ,A1,11H IN COLUMN ,A1)            S0910010
0050           1002 CALL ERRPRT (ERROR)                                           S0910020
0051                WRITE (6,2000) A0,D1                                          S0910030
0052           2002 WRITE (6,3002) A3,A4,A5,A6,N2,(B(K),C(K),K=1,7),A7            S0910040
0053           3002 FORMAT(1H ,4A1,1X,I1,7(A7,A3),3X,A1)                          S0910043
0054                N2=0                                                          S0910045
0055                GO TO 130                                                     S0910050
```

```
0056      1003 CALL ERRPRT (ERROR)                                                  S0910060
0057           WRITE (6,2000) A1,D2                                                 S0910070
0058           GO TO 2002                                                           S0910080
0059      1004 CALL ERRPRT (ERROR)                                                  S0910150
0060           WRITE (6,2000) A2,D3                                                 S0910160
0061           GO TO 2002                                                           S0910170
0062      1005 CALL ERRPRT (ERROR)                                                  S0910180
0063           WRITE (6,2000) D,D4                                                  S0910190
0064           GO TO 2002                                                           S0910200
0065      1006 CALL ERRPRT (ERROR)                                                  S0910210
0066           IP1=I1-1                                                             S0910220
0067           WRITE (6,2006) N2,IP1,MFILE                                          S0910230
0068      2006 FORMAT(31H NUMBER OF FILES IN STATEMENT (,I2,43H) PLUS NUMBER OF FS0910240
              1ILES PREVIOUSLY DEFINED (,I2,27H) EXCEEDS PROGRAM MAXIMUM (,I2,1H)S0910250
              2)                                                                    S0910260
0069           GO TO 2002                                                           S0910270
0070      1007 CALL ERRPRT (ERROR)                                                  S0910280
0071           WRITE (6,2007) N2                                                    S0910290
0072      2007 FORMAT(18H NUMBER OF FILES (,I2,34H) IN COLUMN 6 EXCEEDS MAXIMUM OS0910300
              1F 7)                                                                 S0910310
0073           GO TO 2002                                                           S0910320
0074      1008 CALL ERRPRT (ERROR)                                                  S0910322
0075           WRITE (6,2008)                                                       S0910324
0076      2008 FORMAT(31H NUMBER OF FILES IS LESS THAN 1)                           S0910326
0077           GO TO 2002                                                           S0910328
0078      1009 CALL ERRPRT (ERROR)                                                  S0910330
0079           WRITE (6,2009) FILNAM(L1)                                            S0910340
0080      2009 FORMAT(21H DUPLICATE FILE NAME ,A7)                                  S0910350
0081           GO TO 2002                                                           S0910360
0082      1011 CALL ERRPRT (ERROR)                                                  S0910370
0083           WRITE (6,2011) A(L1)                                                 S0910380
0084      2011 FORMAT(13H UNIT NUMBER ,A3,31H NOT DEFINED BY AN ID STATEMENT)       S0910390
0085           GO TO 2002                                                           S0910400
0086      1121 CALL ERRPRT (ERROR)                                                  S0910410
0087           WRITE (6,2121) A(L1)                                                 S0910420
0088      2121 FORMAT(6H UNIT ,A3,27H ALREADY ASSIGNED TO A FILE)                   S0910430
0089           GO TO 2002                                                           S0910440
0090           END                                                                  S0999999
```

SCALAR MAP

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
BLANK8	1C0	I1	1C8	J1	1CC	MFILE	1D0	A3	1D4
A4	1D8	A5	1DC	A6	1E0	N2	1E4	K	1E8
A7	1EC	A0	1F0	A1	1F4	A2	1F8	D	1FC
K1	200	L1	204	K2	208	L2	20C	L3	210
MCUU	214	K3	218	L4	21C	L5	220	BLANK4	224
ERROR	228	D1	22C	D2	230	D3	234	D4	238
IP1	23C								

ARRAY MAP

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
FILPNT	240	CUUPNT	244	CUU	248	DEVCLS	24C	A	250
C	2A0	FILNAM	2BC	B	2C0				

SUBPROGRAMS CALLED

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
IBCOM=	2F8	ERRPRT	2FC	MINO	300				

FORMAT STATEMENT MAP

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
2	33C	2000	351	3002	378	2006	390	2007	406
2008	442	2009	465	2011	480	2121	484		

TOTAL MEMORY REQUIREMENTS 000052 BYTES

```
0001          SUBROUTINE FILEIO(A,D,H,I,O,T,U,D1,D2,BLANK8,A0,A1,A2,MFILE,      S1000010
             1FILNAM,FILPNT,CUUPNT,DEVCLS,DEVLSS,DEVSRW,VOLUME,RCDSIZ,BLKFCT,    S1000020
             2MULT,LTNCY,SEEK,FACTOR,IOPNT,NAME,EIGHTY,ERROR,ERRORX,CLASS,       S1000030
             3DEVICE,D3,D4,D5)                                                   S1000035
      C ***********************************************************************
      C
      C READ AND PROCESS JBIO AND JFIO STATEMENTS
      C
      C ***********************************************************************
0002          REAL I,O,MULT,LTNCY                                               S1000036
0003          INTEGER FILPNT(20),CUUPNT(20),ERROR,ERRORX,DEVICE                 S1000037
0004          DIMENSION DEVCLS(10),DEVLSS(10),DEVSRW(10),IFIXIT(6)              S1000038
0005          DOUBLE PRECISION FILNAM(20),NAME,BLANK8,A8                        S1000039
      C
      C READ AN IFIO, JBIO, OR JFIO STATEMENT
      C
0006          READ (5,1) A3,A4,A5,A6,A7,A8,VOLUME,RCDSIZ,BLKFCT,MULT,A9,LTNCY,  S1000040
             1SEEK,FACTOR,EIGHTY                                                 S1000050
0007        1 FORMAT(5A1,A7,F8.0,F5.0,F2.0,2A1,2F3.0,F4.3,40X,A1)              S1000060
0008          INAME=1                                                           S1000065
0009          IF(A3-A0)1001,2,1001                                              S1000070
0010        2 IF(A4-A1)1002,6,1002                                              S1000080
0011        6 IF(A5-A2)1006,7,1006                                              S1000120
0012        7 IF(A6-I)1007,8,1007                                               S1000140
0013        8 IF(A7-O)1008,10,1008                                              S1000150
      C
      C CHECK FOR VALID FILE NAME
      C
0014       10 DO 11 I1=1,MFILE                                                  S1000160
0015          IF(FILNAM(I1)-A8)11,12,11                                         S1000170
0016       11 CONTINUE                                                          S1000180
0017          GO TO 1011                                                        S1000190
      C
      C SAVE FILE NAME AND POINTERS TO UNIT NUMBER AND DEVICE TYPE AND MODEL NUMBER
      C
0018       12 INAME=2                                                           S1000195
0019          NAME=FILNAM(I1)                                                   S1000200
0020          FILNAM(I1)=BLANK8                                                 S1000210
0021          IOPNT=FILPNT(I1)                                                  S1000220
0022          J1=CUUPNT(IOPNT)                                                  S1000230
0023          DEVICE=J1                                                         S1000235
      C
      C CHECK FOR VALID DEVICE CLASS
      C
0024          IF(A9-A)13,17,13                                                  S1000240
0025       13 IF(A9-D)14,21,14                                                  S1000250
0026       14 IF(A9-H)15,19,15                                                  S1000260
0027       15 IF(A9-T)16,22,16                                                  S1000270
```

```
0028          16 IF(A9-U)1016,23,1016                                          S1000280
0029          17 CLASS=DEVCLS(J1)                                              S1000290
0030             IF(CLASS-D)24,210,24                                          S1000300
0031          19 CLASS=DEVCLS(J1)                                              S1000310
0032             IF(CLASS-D)20,210,20                                          S1000320
0033          20 IF(CLASS-T)1020,24,1020                                       S1000330
0034          21 CLASS=DEVCLS(J1)                                              S1000340
0035             IF(CLASS-D)1020,210,1020                                      S1000350
            C
            C SAVE LATENCY TIME FOR DISK DEVICE
            C
0036         210 IF(LTNCY-0.)1210,211,212                                      S1000360
0037         211 LTNCY=DEVLSS(J1)                                              S1000370
            C
            C SAVE SEEK TIME FOR DISK DEVICE
            C
0038         212 IF(SEEK-0.)1212,213,24                                        S1000380
0039         213 SEEK=DEVSRW(J1)                                               S1000390
0040             RETURN                                                        S1000400
0041          22 CLASS=DEVCLS(J1)                                              S1000410
0042             IF(CLASS-T)1020,24,1020                                       S1000420
0043          23 CLASS=DEVCLS(J1)                                              S1000430
0044             IF(CLASS-U)1020,24,1020                                       S1000440
0045          24 RETURN                                                        S1000450
            C ************************************************************************
            C
            C ERROR MESSAGES
            C
            C ************************************************************************
0046        2000 FORMAT(18H PROGRAM EXPECTED ,A1,11H IN COLUMN ,A1)            S1010010
0047        1001 CALL ERRPRT (ERROR)                                           S1010020
0048             WRITE (6,2000) A0,D1                                          S1010030
0049        2001 IFIXIT(1)=VOLUME                                              S1010040
0050             IFIXIT(2)=RCDSIZ                                              S1010042
0051             IFIXIT(3)=BLKFCT                                              S1010044
0052             IFIXIT(4)=LTNCY                                               S1010046
0053             IFIXIT(5)=SEEK                                                S1010048
0054             IFIXIT(6)=1000*FACTOR                                         S1010050
0055             WRITE (6,3001) A3,A4,A5,A6,A7,A8,IFIXIT(1),IFIXIT(2),IFIXIT(3),  S1010052
                1MULT,A9,IFIXIT(4),IFIXIT(5),IFIXIT(6),EIGHTY                  S1010054
0056        3001 FORMAT(1H ,5A1,A7,I8,I5,I2,2A1,2I3,I4,40X,A1)                 S1010056
0057             GO TO (3003,3002),INAME                                       S1010057
0058        3002 FILNAM(I1)=NAME                                               S1010058
0059        3003 ERRORX=ERRORX+1                                               S1010060
0060             RETURN                                                        S1010070
0061        1002 CALL ERRPRT (ERROR)                                           S1010080
0062             WRITE (6,2000) A1,D2                                          S1010090
0063             GO TO 2001                                                    S1010100
```

```
0064        1006 CALL ERRPRT (ERROR)                                              S1010170
0065             WRITE (6,2000) A2,D3                                             S1010180
0066             GO TO 2001                                                       S1010190
0067        1007 CALL ERRPRT (ERROR)                                              S1010200
0068             WRITE (6,2000) I,D4                                              S1010210
0069             GO TO 2001                                                       S1010220
0070        1008 CALL ERRPRT (ERROR)                                              S1010230
0071             WRITE (6,2000) O,D5                                              S1010240
0072             GO TO 2001                                                       S1010250
0073        1011 CALL ERRPRT (ERROR)                                              S1010260
0074             WRITE (6,2011) A8,A3,A4,A5,D,A3,A4,A5,I,O                        S1010270
0075        2011 FORMAT(6H FILE ,A7,1X,16HNOT DEFINED BY A,1X,4A1,1X,33HSTATEMENT OS1010280
                1R APPEARED IN PREVIOUS,1X,5A1,1X,9HSTATEMENT)                    S1010285
0076             GO TO 2001                                                       S1010290
0077        1016 CALL ERRPRT (ERROR)                                              S1010300
0078             WRITE (6,2016)                                                   S1010310
0079        2016 FORMAT(51H DEVICE CLASS IN COLUMN 29 MUST BE A, D, H, T, OR U)   S1010320
0080             GO TO 2001                                                       S1010330
0081        1020 CALL ERRPRT (ERROR)                                              S1010340
0082             WRITE (6,2020)                                                   S1010350
0083        2020 FORMAT(80H DEVICE CLASS OF THIS STATEMENT DOES NOT AGREE WITH DEVIS1010360
                1CE CLASS OF HD STATEMENT)                                        S1010370
0084             GO TO 2001                                                       S1010380
0085        1210 CALL ERRPRT (ERROR)                                              S1010390
0086             WRITE (6,2210)                                                   S1010400
0087        2210 FORMAT(51H LATENCY TIME IN COLUMNS 30-32 MUST BE NON-NEGATIVE)   S1010410
0088             GO TO 2001                                                       S1010420
0089        1212 CALL ERRPRT (ERROR)                                              S1010430
0090             WRITE (6,2112)                                                   S1010440
0091        2112 FORMAT(48H SEEK TIME IN COLUMNS 33-35 MUST BE NON-NEGATIVE)      S1010450
0092             GO TO 2001                                                       S1010460
0093             END                                                              S1099999
```

SCALAR MAP

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
A3	1B0	NAME	1B8	BLANK8	1C0	A3	1C8	A4	1CC
A5	1D0	A6	1D4	A7	1D8	VOLUME	1DC	RCDSIZ	1E0
BLKFCI	1E4	MULT	1E8	A9	1EC	LTNCY	1F0	SEEK	1F4
FACTOR	1F8	EIGHTY	1FC	INAME	200	A0	204	A1	208
A2	20C	I	210	O	214	I1	218	MFILE	21C
IOPNT	220	J1	224	DEVICE	228	A	22C	D	230
H	234	T	238	U	23C	CLASS	240	ERROR	244
D1	248	ERRORX	24C	D2	250	D3	254	D4	258
D5	25C								

ARRAY MAP

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
FILPNT	260	CUUPNT	264	DEVCLS	268	DEVLSS	26C	DEVSRW	270
IFIXIT	274	FILNAM	28C						

SUBPROGRAMS CALLED

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
IBCOM=	290	ERRPRT	294						

FORMAT STATEMENT MAP

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
1	2C0	2000	2E1	3001	308	2011	327	2016	385
2020	3BC	2210	410	2112	447				

TOTAL MEMORY REQUIREMENTS 000ECE BYTES

```
0001          SUBROUTINE TOD(WEEK,DON,HON,MON,DOFF,HOFF,MOFF,HHON,MMON,HHOFF,   S1100010
             1MMOFF,TCASE,DAY,ERROR,ERRORX)                                      S1100020
        C ***********************************************************************
        C
        C CHECK BEGIN AND END TIMES FOR PROGRAM
        C
        C ***********************************************************************
0002          INTEGER DON,HON,DOFF,HOFF,HHON(7),HHOFF(7),TCASE,ERROR,WEEK,ERRORXS1100023
0003          DIMENSION MMON(7),MMOFF(7)                                         S1100026
0004          TCASE=1                                                            S1100030
        C
        C IS IT A DAILY PROGRAM OR A WEEKLY PROGRAM
        C
0005          IF(DAY-1)100,50,50                                                 S1100033
        C
        C DAILY PROGRAM.  COMPARE DAY ON AND DAY OFF WITH DAY IN COLUMN 2
        C
0006       50 IF(DAY-DON)1050,51,1050                                            S1100035
0007       51 IF(DAY-DOFF)1051,4,1051                                            S1100037
        C
        C WEEKLY PROGRAM.  CHECK FOR VALID DAY ON AND DAY OFF
        C
0008      100 IF(DON-1)1000,1,1                                                  S1100040
0009        1 IF(DON-WEEK)2,2,1000                                               S1100050
0010        2 IF(DOFF-1)1002,3,3                                                 S1100060
0011        3 IF(DOFF-WEEK)4,4,1002                                              S1100070
        C
        C CHECK TIMES FOR VALIDITY AND CONSISTENCY AND DETERMINE PROGRAM TYPE
        C
0012        4 IF(HON-23)45,45,1004                                               S1100080
0013       45 IF(HON-0)1004,5,5                                                  S1100085
0014        5 IF(HOFF-23)55,55,1005                                              S1100090
0015       55 IF(HOFF-0)1005,6,6                                                 S1100095
0016        6 IF(MON-59)65,65,1006                                               S1100100
0017       65 IF(MON-0)1006,7,7                                                  S1100105
0018        7 IF(MOFF-59)75,75,1007                                              S1100110
0019       75 IF(MOFF-0)1007,8,8                                                 S1100115
0020        8 IF(DON-DOFF)11,9,1008                                              S1100120
0021        9 IF(HON-HOFF)12,10,1008                                             S1100130
0022       10 IF(MON-MOFF)12,1008,1008                                           S1100140
        C
        C WEFKLY PROGRAM
        C
0023       11 TCASE=3                                                            S1100150
0024       12 IF(HON-HHON(DON))1004,13,125                                       S1100160
0025       13 IF(MON-MMON(DON))1006,15,125                                       S1100170
0026      125 IF(HON-HHOFF(DON))14,135,1004                                      S1100173
0027      135 IF(MON-MMOFF(DON))14,1006,1006                                     S1100176
```

```
            C
            C NOT FULL WORKING DAY PROGRAM
            C
 0028          14 TCASE=MAX0(TCASE,2)                                          S1100180
 0029          15 IF(HOFF-HHOFF(DOFF))155,16,1005                              S1100190
 0030          16 IF(MOFF-MMOFF(DOFF))155,18,1007                              S1100200
 0031         155 IF(HOFF-HHON(DOFF))1005,165,17                               S1100203
 0032         165 IF(MOFF-MMON(DOFF))1007,1007,17                              S1100206
            C
            C NOT FULL WORKING DAY PROGRAM
            C
 0033          17 TCASE=MAX0(TCASE,2)                                          S1100210
 0034          18 RETURN                                                       S1100220
            C ************************************************************************
            C
            C ERROR MESSAGES
            C
            C ************************************************************************
 0035        1000 CALL ERRPRT (ERROR)                                          S1110010
 0036             WRITE (6,2000)                                               S1110020
 0037        2000 FORMAT(18H DAY ON IS INVALID)                                S1110030
 0038        2001 ERRORX=ERRORX+1                                              S1110040
 0039             RETURN                                                       S1110050
 0040        1002 CALL ERRPRT (ERROR)                                          S1110060
 0041             WRITE (6,2002)                                               S1110070
 0042        2002 FORMAT(19H DAY OFF IS INVALID)                               S1110080
 0043             GO TO 2001                                                   S1110090
 0044        1004 CALL ERRPRT (ERROR)                                          S1110100
 0045             WRITE (6,2004)                                               S1110110
 0046        2004 FORMAT(19H HOUR ON IS INVALID)                               S1110120
 0047             GO TO 2001                                                   S1110130
 0048        1005 CALL ERRPRT (ERROR)                                          S1110140
 0049             WRITE (6,2005)                                               S1110150
 0050        2005 FORMAT(20H HOUR OFF IS INVALID)                              S1110160
 0051             GO TO 2001                                                   S1110170
 0052        1006 CALL ERRPRT (ERROR)                                          S1110180
 0053             WRITE (6,2006)                                               S1110190
 0054        2006 FORMAT(21H MINUTE ON IS INVALID)                             S1110200
 0055             GO TO 2001                                                   S1110210
 0056        1007 CALL ERRPRT (ERROR)                                          S1110220
 0057             WRITE (6,2007)                                               S1110230
 0058        2007 FORMAT(22H MINUTE OFF IS INVALID)                            S1110240
 0059             GO TO 2001                                                   S1110250
 0060        1008 CALL ERRPRT (ERROR)                                          S1110260
 0061             WRITE (6,2008)                                               S1110270
 0062        2008 FORMAT(42H TIME OFF IS LESS THAN OR EQUAL TO TIME ON)        S1110280
 0063             GO TO 2001                                                   S1110290
 0064        1050 CALL ERRPRT (ERROR)                                          S1110300
```

```
            WRITE (6,2050)                                          S1110310
       2050 FORMAT(43H DAY ON DOES NOT AGREE WITH DAY IN COLUMN 2)  S1110320
            GO TO 2001                                              S1110330
       1051 CALL ERRPRT (ERROR)                                     S1110340
            WRITE (6,2051)                                          S1110350
       2051 FORMAT(44H DAY OFF DOES NOT AGREE WITH DAY IN COLUMN 2) S1110360
            GO TO 2001                                              S1110370
            END                                                     S1199999
```

SCALAR MAP

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
TCASE	1C8	DAY	1CC	DON	1D0	DOFF	1D4	WEEK	1D8
HON	1DC	HOFF	1E0	MON	1E4	MOFF	1E8	ERROR	1EC
ERRORX	1F0								

ARRAY MAP

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
HHON	1F4	HHOFF	1F8	MMON	1FC	MMOFF	200		

SUBPROGRAMS CALLED

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
ERRPRT	204	IBCOM=	208	MAXO	20C				

FORMAT STATEMENT MAP

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
2000	244	2002	25A	2004	271	2005	288	2006	2A0
2007	2B9	2008	2D3	2050	301	2051	330		

TOTAL MEMORY REQUIREMENTS 000942 BYTES

```
0001          SUBROUTINE SWITCH(JOBNM1,STPNM1,DON1,HON1,MON1,DOFF1,HOFF1,MOFF1, S1200010
             1JOBNM2,STPNM2,DON2,HON2,MON2,DOFF2,HOFF2,MOFF2)                  S1200020
        C ***********************************************************************
        C
        C MOVE PROGRAM DESCRIPTIONS TO FIRST PLACE IN TABLE FOR CURRENT PROGRAM TYPE
        C
        C ***********************************************************************
0002          DOUBLE PRECISION JOBNM1,JOBNM2,STPNM1,STPNM2                     S1200030
0003          INTEGER DON1,DON2,HON1,HON2,DOFF1,DOFF2,HOFF1,HOFF2              S1200040
0004          JOBNM2=JOBNM1                                                    S1200050
0005          STPNM2=STPNM1                                                    S1200060
0006          DON2=DON1                                                        S1200070
0007          HON2=HON1                                                        S1200080
0008          MON2=MON1                                                        S1200090
0009          DOFF2=DOFF1                                                      S1200100
0010          HOFF2=HOFF1                                                      S1200110
0011          MOFF2=MOFF1                                                      S1200120
0012          RETURN                                                           S1200130
0013          END                                                              S1200140
```

SCALAR MAP

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
JOBNM2	90	JOBNM1	98	STPNM2	A0	STPNM1	A8	DON2	B0
DON1	84	HON2	88	HON1	BC	MON2	C0	MON1	C4
DOFF2	C8	DOFF1	CC	HOFF2	D0	HOFF1	D4	MOFF2	D8
MOFF1	DC								

TOTAL MEMORY REQUIREMENTS 000286 BYTES

```
0001          SUBROUTINE JOHNSN(A,B,N,IN)                                        S1300010
         C ***********************************************************************
         C
         C SOLVE TWO-MACHINE TWO-OPERATION SEQUENCING PROBLEM
         C
         C ***********************************************************************
0002          DIMENSION A(5),B(5),N(5)                                           S1300020
0003          I=1                                                                S1300030
0004          K=1                                                                S1300040
0005          L=IN                                                               S1300050
         C
         C WHICH IS SMALLER -- FIRST OPERATION OR SECOND OPERATION
         C
0006        5 IF(A(I)-B(I))20,20,10                                              S1300060
         C
         C SECOND OPERATION TIME.  COMPARE THIS TIME WITH THE OTHER JOBS' TIMES
         C
0007       10 J=I                                                                S1300070
0008          I=I+1                                                              S1300080
         C
         C IS THERE ANOTHER JOB WHICH HAS NOT BEEN SEQUENCED
         C
0009       11 DO 15 IY=I,IN                                                      S1300090
0010          I=IY                                                               S1300100
0011          DO 16 IZ=1,IN                                                      S1300110
0012          IF(I-N(IZ))16,15,16                                                S1300120
0013       16 CONTINUE                                                           S1300130
         C
         C YES.  COMPARE OPERATION TIMES WITH THAT JOB
         C
0014          GO TO 30                                                           S1300140
0015       15 CONTINUE                                                           S1300150
         C
         C NO.  PLACE THIS JOB IN THE SEQUENCE
         C
0016          GO TO 75                                                           S1300160
         C
         C IS EITHER OF THE SECOND JOB'S TIMES SMALLER THAN THE FIRST JOB'S TIME
         C
0017       30 IF(A(I)-B(I))35,35,36                                              S1300170
0018       35 IF(A(I)-B(J))60,60,70                                              S1300180
0019       36 IF(B(J)-B(I))70,70,50                                              S1300190
         C
         C YES.  SECOND OPERATION TIME IS SMALLER.  USE THIS TIME TO COMPARE WITH
         C
0020       50 J=I                                                                S1300200
         C
         C ANY MORE JOBS WHICH HAVE NOT BEEN SEQUENCED
```

```
              C
0021             70 I=I+1                                                   S1300210
0022                IF(I-IN)11,11,75                                        S1300220
              C
              C NO.  SEQUENCE THIS JOB BY PLACING IT IN THE LAST UNFILLED SPACE
              C
0023             75 N(L)=J                                                  S1300230
0024                L=L-1                                                   S1300240
0025                GO TO 90                                                S1300250
              C
              C FIRST OPERATION IS SMALLER.  COMPARE THIS TIME WITH THE OTHER JOBS' TIMES
              C
0026             20 J=I                                                     S1300260
0027                I=I+1                                                   S1300270
              C
              C IS THERE ANOTHER JOB WHICH HAS NOT BEEN SEQUENCED
              C
0028             21 DO 25 IY=I,IN                                           S1300280
0029                I=IY                                                    S1300290
0030                DO 26 IZ=1,IN                                           S1300300
0031                IF(I-N(IZ))26,25,26                                     S1300310
0032             26 CONTINUE                                                S1300320
              C
              C YES.  COMPARE OPERATION TIMES WITH THAT JOB
              C
0033                GO TO 40                                                S1300330
0034             25 CONTINUE                                                S1300340
              C
              C NO.  PLACE THIS JOB IN THE SEQUENCE
              C
0035                GO TO 85                                                S1300350
              C
              C IS EITHER OF THE SECOND JOB'S TIMES SMALLER THAN THE FIRST JOB'S TIME
              C
0036             40 IF(A(I)-B(I))45,45,46                                   S1300360
0037             45 IF(A(J)-A(I))80,80,60                                   S1300370
0038             46 IF(A(J)-B(I))80,80,50                                   S1300380
              C
              C YES.  FIRST OPERATION TIME IS SMALLER.  USE THIS TIME TO COMPARE WITH
              C
0039             60 J=I                                                     S1300390
              C
              C ANY MORE JOBS WHICH HAVE NOT BEEN SEQUENCED
              C
0040             80 I=I+1                                                   S1300400
0041                IF(I-IN)21,21,85                                        S1300410
              C
              C NO.  SEQUENCE THIS JOB BY PLACING IT IN THE FIRST UNFILLED SPACE
```

```
                C                                                              
0042               85 N(K)=J                                                   S1300420
0043                  K=K+1                                                    S1300430
                C
                C FIND THE FIRST JOB WHICH HAS NOT BEEN SEQUENCED
                C
0044               90 DO 95 IY=1,IN                                            S1300440
0045                  I=IY                                                     S1300450
0046                  DO 96 IZ=1,IN                                            S1300460
0047                  IF(I-N(IZ))96,95,96                                      S1300470
0048               96 CONTINUE                                                 S1300480
0049                  GO TO 98                                                 S1300490
0050               95 CONTINUE                                                 S1300500
                C
                C IS IT THE ONLY JOB WHICH HAS NOT BEEN SEQUENCED
                C
0051               98 IF(K+1-L)5,5,99                                          S1300510
                C
                C YES.  PLACE THAT JOB IN THE ONLY REMAINING SPACE IN THE TABLE
                C
0052               99 N(K)=I                                                   S1300520
0053                  RETURN                                                   S1300530
0054                  END                                                      S1300540
```

SCALAR MAP

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
I	173	K	17C	L	180	IN	184	J	188
IY	18C	IZ	190						

ARRAY MAP

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
A	194	B	198	N	19C				

TOTAL MEMORY REQUIREMENTS 0005A6 BYTES

```
0001         SUBROUTINE IGNSCH(A,B,C,D,DX,DP,DPX,L,N,IN,CASE,ERROR,ERRORX)     S1400010
           C ***************************************************************************
           C
           C SOLVE THREE-MACHINE THREE-OPERATION SEQUENCING PROBLEM
           C
           C ***************************************************************************
0002         INTEGER CASE,ERROR,ERRORX,PL(5),POPER(120),POPERR,POPERO            S1400015
0003         DIMENSION A(5),B(5),C(5),D(5),DX(5),DP(5),DPX(5),L(5),N(5),M(5)     S1400020
0004         DIMENSION BOUND(120),LIST(120),TIMEA(120),TIMEB(120),TIMEC(120)     S1400023
0005         DIMENSION PA(5),PB(5),PD(5),PDX(5),PDP(5),PDPX(5)                   S1400026
0006         NLIST=120                                                           S1400030
           C
           C ARE THERE PARALLEL OPERATIONS ON INPUT
           C
0007         GO TO (1,1,103,1,105,1,1),CASE                                      S1400031
           C
           C YES.  SAME OPERATION, SAME MACHINE.  FIND OPTIMAL INPUT SEQUENCE
           C
0008     103 CALL PARAL1 (D,DP,A,B,L,IN)                                         S1400032
0009         GO TO 1                                                             S1400033
           C
           C YES.  SAME OPERATION, DIFFERENT MACHINES. .FIND OPTIMAL INPUT SEQUENCE
           C
0010     105 CALL PARAL2(D,DX,DP,DPX,A,B,L,IN)                                   S1400036
           C
           C INITIALIZE SUBROUTINE
           C
0011       1 JN=IN-1                                                             S1400040
0012         NEXT=1                                                              S1400050
0013         LIST(1)=0                                                           S1400060
0014         BOUND(1)=0.                                                         S1400070
0015         TIMEA(1)=0.                                                         S1400080
0016         TIMEB(1)=0.                                                         S1400090
0017         TIMEC(1)=0.                                                         S1400100
0018         DO 100 I=1,NLIST                                                    S1400102
0019     100 POPER(I)=0                                                          S1400103
0020         POPERR=0                                                            S1400104
0021         DO 101 I=1,IN                                                       S1400105
0022     101 N(I)=0                                                              S1400106
           C
           C DETERMINE THE JOB SEQUENCE FOR THE FIRST NODE ON THE LIST
           C
0023       2 DO 3 I=1,IN                                                         S1400110
0024       3 M(I)=I                                                              S1400120
0025         GO TO (60,40),NEXT                                                  S1400130
0026      40 DO 4 I=1,JN                                                         S1400140
0027       4 N(I)=LIST(1)/(10**(JN-I))                                           S1400150
0028         J=JN-1                                                              S1400160
```

```
0029           DO 5 I=1,J                                          S1400170
0030           K=JN-I                                              S1400180
0031         5 N(K+1)=MOD(N(K+1),10*N(K))                          S1400190
      C
      C ARE THERE PARALLEL OPERATIONS ON OUTPUT
      C
0032        45 GO TO (60,60,60,50,60,50,60),CASE                   S1400191
      C
      C YES.  SAME OPERATION.  SAVE OPTIMAL OUTPUT SEQUENCE
      C
0033        50 DO 52 I=1,IN                                        S1400194
0034        52 L(I)=POPER(I)/(10**(IN-I))                          S1400195
0035           DO 53 I=1,JN                                        S1400197
0036           K=IN-I                                              S1400198
0037        53 L(K+1)=MOD(L(K+1),10*L(K))                          S1400199
0038           GO TO (62,62,62,62),CASE                            S1400209
0039           DO 61 I=1,IN                                        S1400210
0040           I1=L(I)                                             S1400211
0041           GO TO (58,59),I1                                    S1400212
0042        58 B(I)=D(I)                                           S1400213
0043           C(I)=DPX(I)                                         S1400214
0044           GO TO 61                                            S1400215
0045        59 B(I)=DP(I)                                          S1400216
0046           C(I)=DX(I)                                          S1400217
0047        61 CONTINUE                                            S1400218
0048           GO TO 60                                            S1400219
0049        62 DO 65 I=1,IN                                        S1400220
0050           I1=L(I)                                             S1400221
0051           GO TO (63,64),I1                                    S1400222
0052        63 B(I)=D(I)                                           S1400223
0053           C(I)=DP(I)                                          S1400224
0054           GO TO 65                                            S1400225
0055        64 B(I)=DP(I)                                          S1400226
0056           C(I)=D(I)                                           S1400227
0057        65 CONTINUE                                            S1400228
      C
      C HAS THIS NODE SCHEDULED ALL JOBS
      C
0058        60 DO 7 I=1,JN                                         S1400229
0059           IF(N(I)-1)10,6,6                                    S1400230
0060         6 J=N(I)                                              S1400231
0061           M(J)=0                                              S1400232
0062         7 CONTINUE                                            S1400240
      C
      C YES.  OPTIMAL SEQUENCE HAS BEEN FOUND
      C
0063           DO 8 I=1,IN                                         S1400250
0064           IF(M(I)-1)8,9,9                                     S1400260
```

```
0065            8 CONTINUE                                                     S1400270
0066            9 N(IN)=I                                                      S1400280
0067              RETURN                                                       S1400290
            C
            C OPTIMAL SEQUENCE HAS NOT BEEN FOUND.  REMOVE FIRST NODE FROM LIST
            C
0068           10 LISTO=LIST(1)                                                S1400300
0069              TIMEAO=TIMEA(1)                                              S1400310
0070              TIMEBO=TIMEB(1)                                              S1400320
0071              TIMECO=TIMEC(1)                                              S1400330
0072              POPERO=POPER(1)                                              S1400335
0073              GO TO (13,11),NEXT                                           S1400340
0074           11 J=NEXT-2                                                     S1400350
            C
            C MOVE ALL NODES UP ONE POSITION IN LIST
            C
0075              DO 12 K=1,J                                                  S1400360
0076              LIST(K)=LIST(K+1)                                            S1400370
0077              BOUND(K)=BOUND(K+1)                                          S1400380
0078              TIMEA(K)=TIMEA(K+1)                                          S1400390
0079              TIMEB(K)=TIMEB(K+1)                                          S1400400
0080              TIMEC(K)=TIMEC(K+1)                                          S1400410
0081           12 POPER(K)=POPER(K+1)                                          S1400415
0082              NEXT=NEXT-1                                                  S1400420
0083           13 DO 21 J=1,IN                                                 S1400430
            C
            C IS THE LIST FILLED
            C
0084              IF(NEXT-NLIST)131,131,22                                     S1400431
            C
            C NO. FIND A JOB WHICH THE JUST REMOVED NODE HAS NOT SCHEDULED
            C
0085          131 IF(M(J)-1)21,14,14                                           S1400440
            C
            C CREATE A NODE FOR THE SEQUENCE CONTAINING THE ADDITIONAL JOB
            C
0086           14 LISTR=LISTO+(J*(10**(JN-I)))                                 S1400445
            C
            C COMPUTE TIMEA, TIMEB, AND TIMEC FOR THE SEQUENCE CONTAINING THE ADDITIONAL JOB
            C
0087              TIMEAR=TIMEAO+A(J)                                           S1400450
0088              ASUM=0.                                                      S1400460
0089              BSUM=0.                                                      S1400470
0090              CSUM=0.                                                      S1400471
            C
            C WHICH THREE-OPERATION CASE IS THIS
            C
0091              GO TO (141,142,141,134,141,134,147),CASE                     S1400472
```

```
C
C PARALLEL OPERATIONS ON OUTPUT.  SAME OPERATION
C
  134 IF(TIMEBO-TIMEAR)135,133,132                                S1400473
  132 IF(TIMECO-TIMEAR)137,148,148                                S1400474
  133 IF(TIMECO-TIMEAR)136,148,148                                S1400475
  135 IF(TIMECO-TIMEAR)136,136,137                                S1400476
  136 I1=1                                                        S1400477
      GO TO 138                                                   S1400478
  137 PD(1)=AMAX1(0.,TIMEBO-TIMEAR)                               S1400479
      PDX(1)=AMAX1(0.,TIMECO-TIMEAR)                              S1400480
      PDP(1)=AMAX1(0.,TIMEBO-TIMEAR)                              S1400481
      PDPX(1)=AMAX1(0.,TIMECO-TIMEAR)                             S1400482
      I1=2                                                        S1400483
  138 J1=I1                                                       S1400484
      IPN=IN-I+I1                                                 S1400485
      DO 140 K=1,IN                                               S1400486
      IF(M(K)-1)140,139,139                                       S1400487
  139 PD(I1)=D(K)                                                 S1400488
      PDX(I1)=DX(K)                                               S1400489
      PDP(I1)=DP(K)                                               S1400490
      PDPX(I1)=DPX(K)                                             S1400491
      I1=I1+1                                                     S1400492
  140 CONTINUE                                                    S1400493
      GO TO (143,143,143,143),CASE                                S1400494
      CALL PARAL2 (PD,PDX,PDP,PDPX,PA,PB,PL,IPN)                  S1400495
      GO TO 144                                                   S1400496
  143 CALL PARAL1 (PD,PDP,PA,PB,PL,IPN)                           S1400497
  144 DO 146 K=1,IN                                               S1400498
      IF(M(K)-1)146,145,145                                       S1400499
  145 B(K)=PA(J1)                                                 S1400500
      C(K)=PB(J1)                                                 S1400501
      L(K)=PL(J1)                                                 S1400502
      J1=J1+1                                                     S1400503
  146 CONTINUE                                                    S1400504
      POPERR=0                                                    S1400505
      DO 149 I1=1,IN                                              S1400506
  149 POPERR=POPERR+(L(I1)*(10**(IN-I1)))                         S1400507
      GO TO 142                                                   S1400508
  148 POPERR=POPERO                                               S1400509
      GO TO 142                                                   S1400510
C
C PARALLEL OPERATIONS ON INPUT
C
  141 TIMEBR=TIMEBO+B(J)                                          S1400512
      TIMECR=AMAX1(TIMEAR,TIMEBR,TIMECO)+C(J)                     S1400513
      GO TO 150                                                   S1400514
C
```

```
              C PARALLEL OPERATIONS ON OUTPUT
              C
0133            142 TIMEBR=AMAX1(TIMEAR,TIMEBO)+B(J)                       S1400520
0134                TIMECR=AMAX1(TIMEAR,TIMECO)+C(J)                       S1400530
0135                BMIN=10.**74                                           S1400531
0136                GO TO 150                                              S1400532
              C
              C NO PARALLEL OPERATIONS
              C
0137            147 TIMEBR=AMAX1(TIMEAR,TIMEBO)+B(J)                       S1400533
0138                TIMECR=AMAX1(TIMEBR,TIMECO)+C(J)                       S1400534
0139                BCMIN=10.**74                                          S1400535
0140            150 CMIN=10.**74                                           S1400536
0141                DO 17 K=1,IN                                           S1400540
0142                IF(M(K)-1)17,15,15                                     S1400550
0143             15 IF(K-J)16,17,16                                        S1400560
0144             16 ASUM=ASUM+A(K)                                         S1400570
0145                BSUM=BSUM+B(K)                                         S1400580
0146                CSUM=CSUM+C(K)                                         S1400590
              C
              C WHICH THREE-OPERATION CASE IS THIS
              C
0147                GO TO (161,162,161,162,161,162,167),CASE               S1400591
              C
              C PARALLEL OPERATIONS ON OUTPUT
              C
0148            162 IF(B(K)-C(K))161,163,164                               S1400592
0149            163 BMIN=AMIN1(BMIN,B(K))                                  S1400593
0150                GO TO 161                                              S1400594
0151            164 BMIN=AMIN1(BMIN,B(K))                                  S1400595
0152                GO TO 17                                               S1400596
              C
              C NO PARALLEL OPERATIONS
              C
0153            167 BCMIN=AMIN1(BCMIN,B(K)+C(K))                           S1400600
              C
              C PARALLEL OPERATIONS ON INPUT
              C
0154            161 CMIN=AMIN1(CMIN,C(K))                                  S1400610
0155             17 CONTINUE                                               S1400620
0156                TIMECT=TIMECR+CSUM                                     S1400650
              C
              C WHICH THREE-OPERATION CASE IS THIS
              C
0157                GO TO (171,172,171,172,171,172,177),CASE               S1400651
              C
              C PARALLEL OPERATIONS ON INPUT
              C
```

```
0158          171 TIMEAT=TIMEAR+ASUM+CMIN                                        S1400652
0159              TIMEBT=TIMEBR+BSUM+CMIN                                        S1400653
0160              GO TO 178                                                      S1400654
            C
            C PARALLEL OPERATIONS ON OUTPUT
            C
0161          172 TIMEAT=TIMEAR+ASUM+AMIN1(BMIN,CMIN)                            S1400655
0162              TIMEBT=TIMEBR+BSUM                                             S1400656
0163              GO TO 178                                                      S1400657
            C
            C NO PARALLEL OPERATIONS
            C
0164          177 TIMEAT=TIMEAR+ASUM+BCMIN                                       S1400658
0165              TIMEBT=TIMEBR+BSUM+CMIN                                        S1400659
            C
            C COMPUTE THE LOWER BOUND FOR THE SEQUENCE CONTAINING THE ADDITIONAL JOB
            C
0166          178 BOUNDR=AMAX1(TIMEAT,TIMEBT,TIMECT)                             S1400660
0167              GO TO (181,179),NEXT                                           S1400670
0168          179 K=NEXT-1                                                       S1400680
            C
            C FIND THE PROPER PLACE IN THE LIST FOR THE NEW NODE
            C
0169              DO 18 J1=1,K                                                   S1400690
0170              IF(BOUNDR-BOUND(J1))19,19,18                                   S1400700
0171           18 CONTINUE                                                       S1400710
            C
            C THE NEW NODE'S LOWER BOUND IS LARGER THAN ALL OTHER NODES' LOWER BOUNDS.
            C PLACE THE NEW NODE IN THE NEXT FREE POSITION IN THE LIST
            C
0172          181 LIST(NEXT)=LISTR                                               S1400720
0173              BOUND(NEXT)=BOUNDR                                             S1400730
0174              TIMEA(NEXT)=TIMEAR                                             S1400740
0175              TIMEB(NEXT)=TIMEBR                                             S1400750
0176              TIMEC(NEXT)=TIMECR                                             S1400760
0177              POPER(NEXT)=POPERR                                             S1400765
0178              NEXT=NEXT+1                                                    S1400770
0179              GO TO 21                                                       S1400780
            C
            C THE NEW NODE'S LOWER BOUND IS EQUAL TO OR LESS THAN SOME OTHER NODE'S LOWER
            C BOUND.  MOVE DOWN THE LIST ALL NODES WITH LOWER BOUNDS GREATER THAN OR EQUAL
            C TO THAT OF THE NEW NODE
            C
0180           19 K=NEXT-J1                                                      S1400790
0181              DO 20 I1=1,K                                                   S1400800
0182              LIST(NEXT+1-I1)=LIST(NEXT-I1)                                  S1400810
0183              BOUND(NEXT+1-I1)=BOUND(NEXT-I1)                                S1400820
0184              TIMEA(NEXT+1-I1)=TIMEA(NEXT-I1)                                S1400830
```

```
0185             TIMEB(NEXT+1-I1)=TIMEB(NEXT-I1)                                  S1400840
0186             TIMEC(NEXT+1-I1)=TIMEC(NEXT-I1)                                  S1400850
0187          20 POPER(NEXT+1-I1)=POPER(NEXT-I1)                                  S1400855
           C
           C INSERT THE NEW NODE IN THE LIST
           C
0188             LIST(J1)=LISTR                                                   S1400860
0189             BOUND(J1)=BOUNDR                                                 S1400870
0190             TIMEA(J1)=TIMEAR                                                 S1400880
0191             TIMEB(J1)=TIMEBR                                                 S1400890
0192             TIMEC(J1)=TIMECR                                                 S1400900
0193             POPER(J1)=POPERR                                                 S1400905
0194             NEXT=NEXT+1                                                      S1400910
0195          21 CONTINUE                                                         S1400920
0196             GO TO 2                                                          S1400930
           C
           C LIST FULL.  ALGORITHM CANNOT CONTINUE DUE TO LACK OF STORAGE
           C
0197          22 DO 30 I=1,NLIST                                                  S1400940
           C
           C DETERMINE THE JOB SEQUENCE FOR A NODE
           C
0198             DO 23 J=1,JN                                                     S1400950
0199          23 N(J)=LIST(I)/(10**(JN-J))                                        S1400960
0200             J=JN-1                                                           S1400970
0201             DO 24 I1=1,J                                                     S1400980
0202             K=JN-I1                                                          S1400990
0203          24 N(K+1)=MOD(N(K+1),10*N(K))                                       S1401000
           C
           C HAS THIS NODE SCHEDULED ALL JOBS
           C
0204             DO 26 K=1,JN                                                     S1401010
0205             IF(N(K)-1)30,25,25                                               S1401020
0206          25 J=N(K)                                                           S1401030
0207             M(J)=0                                                           S1401040
0208          26 CONTINUE                                                         S1401050
           C
           C YES.  IS THIS NODE THE FIRST NODE ON THE LIST
           C
0209             IX=2                                                             S1401060
0210             GO TO (45,27),I                                                  S1401070
           C
           C NO.  THE SEQUENCE MAY NOT BE OPTIMAL
           C
0211          27 GO TO 1038                                                       S1401080
0212          30 CONTINUE                                                         S1401090
           C
           C NO NODE HAS SCHEDULED ALL THE JOBS.  AN OPTIMAL SEQUENCE COULD NOT BE FOUND
```

```
          C                                                                                 
 0213           IX=1                                                                S1401100
 0214           GO TO 1030                                                          S1401110
          C ****************************************************************************
          C
          C ERROR MESSAGES
          C
          C ****************************************************************************
 0215      1030 CALL ERRPRT (ERROR)                                                 S1410010
 0216           WRITE (6,2030)                                                      S1410020
 0217      2030 FORMAT(85H SCHEDULING ALGORITHM COULD NOT PRODUCE OPTIMAL DAILY SCS1410030
               1HEDULE DUE TO LACK OF STORAGE)                                      S1410040
 0218           WRITE (6,2031)                                                      S1410050
 0219      2031 FORMAT(41H NO NODE HAS SCHEDULED ALL THE DAILY JOBS)                S1410060
 0220      1032 WRITE (6,2032) NLIST                                                S1410070
 0221      2032 FORMAT(40H NUMBER OF NODES AVAILABLE TO ALGORITHM=,I6)              S1410080
 0222           WRITE (6,2033) IN                                                   S1410090
 0223      2033 FORMAT(32H NUMBER OF JOBS TO BE SCHEDULED=,I2)                      S1410100
 0224           AI=(.5*IN*(IN-1))+1                                                 S1410110
 0225           I=AI                                                                S1410120
 0226           WRITE (6,2034) I                                                    S1410130
 0227      2034 FORMAT(34H MINIMUM NUMBER OF NODES REQUIRED=,I4)                    S1410140
 0228           K=IN                                                                S1410150
 0229           DO 2035 I=2,JN                                                      S1410160
 0230      2035 K=K*I                                                               S1410170
 0231           WRITE (6,2036) K                                                    S1410180
 0232      2036 FORMAT(34H MAXIMUM NUMBER OF NODES REQUIRED=,I7)                    S1410190
 0233           ERRORX=ERRORX+1                                                     S1410200
 0234           GO TO (1037,45),IX                                                  S1410210
 0235      1037 RETURN                                                              S1410220
 0236      1038 CALL ERRPRT (ERROR)                                                 S1410230
 0237           WRITE (6,2030)                                                      S1410240
 0238           WRITE (6,2038) I                                                    S1410250
 0239      2038 FORMAT(5H NODE,1X,I6,52H IS THE FIRST NODE THAT HAS SCHEDULED ALL S1410260
               1DAILY JOBS)                                                         S1410270
 0240           GO TO 1032                                                          S1410280
 0241           END                                                                 S1499999
```

SCALAR MAP

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
NLIST	304	CASE	308	IN	30C	JN	310	NEXT	314
I	318	POPERR	31C	J	320	K	324	I1	328
LISTO	32C	TIMEAO	330	TIMEBO	334	TIMECO	338	POPERO	33C
LISTR	340	TIMEAR	344	ASUM	348	BSUM	34C	CSUM	350
J1	354	IPN	358	TIMEBR	35C	TIMECR	360	BMIN	364
BCMIN	368	CMIN	36C	TIMECT	370	TIMEAT	374	TIMEBT	378
BOUNDR	37C	IX	380	ERROR	384	AI	388	ERRORX	38C

ARRAY MAP

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
PL	390	POPER	3A4	A	584	B	588	C	58C
D	590	DX	594	DP	598	DPX	59C	L	5A0
N	5A4	M	5A8	BOUND	5BC	LISY	79C	TIMEA	97C
TIMEB	B5C	TIMEC	D3C	PA	F1C	PB	F30	PD	F44
PDX	F58	PDP	F6C	PDPX	F80				

SUBPROGRAMS CALLED

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
PARAL1	F94	PARAL2	F98	FIXPI=	F9C	FRXPI=	FA0	ERRPRT	FA4
IBCOM=	FA8	AMAX1	FAC	AMIN1	FB0				

FORMAT STATEMENT MAP

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
2030	1104	2031	115D	2032	118A	2033	11B8	2034	11DE
2036	1206	2038	122E						

TOTAL MEMORY REQUIREMENTS 00288A BYTES

```
0001          SUBROUTINE PARALI(A,B,AX,BX,L,IN)                                S1500010
        C ***********************************************************************
        C
        C SOLVE PARALLEL OPERATION SEQUENCING PROBLEM FOR SAME OPERATION, SAME MACHINES
        C
        C ***********************************************************************
0002          DIMENSION A(5),B(5),AX(5),BX(5),L(5),LMAX(5),LMIN(5)             S1500020
0003          JN=IN-1                                                          S1500030
0004          DO 3 I=1,IN                                                      S1500040
0005          IF(A(I)-B(I))1,1,2                                               S1500050
0006        1 LMAX(I)=2                                                        S1500060
0007          LMIN(I)=1                                                        S1500070
0008          GO TO 3                                                          S1500080
0009        2 LMAX(I)=1                                                        S1500090
0010          LMIN(I)=2                                                        S1500100
0011        3 CONTINUE                                                         S1500110
        C
        C CREATE A NODE FOR EACH TREE
        C
0012          TIMEAX=A(1)                                                      S1500120
0013          TIMEBX=B(1)                                                      S1500130
0014          TIMEAR=A(1)                                                      S1500140
0015          TIMEBR=B(1)                                                      S1500150
0016          L(1)=1                                                           S1500160
0017          I1=LMAX(1)                                                       S1500170
        C
        C WHICH MACHINE HAS THE LARGER OPERATION TIME FOR THE FIRST JOB
        C
0018          GO TO (4,6),I1                                                   S1500180
        C
        C MACHINE A.  ASSIGN SMALLER OPERATION TIMES OF REMAINING JOBS TO MACHINE A
        C
0019        4 DO 5 I=2,IN                                                      S1500190
0020          TIMEAR=TIMEAR+AMIN1(A(I),B(I))                                   S1500200
0021          TIMEBR=TIMEBR+AMAX1(A(I),B(I))                                   S1500210
0022        5 L(I)=LMIN(I)                                                     S1500220
0023          GO TO 8                                                          S1500230
        C
        C MACHINE B.  ASSIGN SMALLER OPERATION TIMES OF REMAINING JOBS TO MACHINE B
        C
0024        6 DO 7 I=2,IN                                                      S1500240
0025          TIMEAR=TIMEAR+AMAX1(A(I),B(I))                                   S1500250
0026          TIMEBR=TIMEBR+AMIN1(A(I),B(I))                                   S1500260
0027        7 L(I)=LMAX(I)                                                     S1500270
        C
        C COMPUTE THE LOWER BOUND FOR THE SEQUENCES CREATED
        C
0028        8 BOUNDO=AMAX1(TIMEAR,TIMEBR)                                      S1500280
```

```
0029            DO 16 I=2,JN                                           S1500290
          C
          C EXTEND THE TREES BY ADDING BOTH POSSIBLE OPERATIONS TO BOTH TREES
          C
0030            TIMEAY=TIMEAX+AMIN1(A(I),B(I))                         S1500300
0031            TIMEBY=TIMEBX+AMAX1(A(I),B(I))                         S1500310
0032            TIMEAX=TIMEAX+AMAX1(A(I),B(I))                         S1500320
0033            TIMEBX=TIMEBX+AMIN1(A(I),B(I))                         S1500330
0034            TIMEAR=TIMEAX                                          S1500340
0035            TIMEBR=TIMEBX                                          S1500350
0036            TIMEAS=TIMEAY                                          S1500360
0037            TIMEBS=TIMEBY                                          S1500370
0038            JX=I+1                                                 S1500380
0039            DO 9 J=JX,IN                                           S1500390
0040            TIMEAR=TIMEAR+AMIN1(A(J),B(J))                         S1500400
0041            TIMEBR=TIMEBR+AMAX1(A(J),B(J))                         S1500410
0042            TIMEAS=TIMEAS+AMAX1(A(J),B(J))                         S1500420
0043          9 TIMEBS=TIMEBS+AMIN1(A(J),B(J))                         S1500430
          C
          C COMPUTE THE LOWER BOUNDS FOR THE NEW SEQUENCES
          C
0044            BOUNDR=AMAX1(TIMEAR,TIMEBR)                            S1500440
0045            BOUNDS=AMAX1(TIMEAS,TIMEBS)                            S1500450
          C
          C THE SMALLER OF THESE TWO LOWER BOUNDS IS THE NEW LOWER BOUND
          C
0046            IF(BOUNDR-BOUNDS)10,10,13                              S1500460
          C
          C IS THE CURRENT LOWER BOUND SMALLER THAN THE NEW LOWER BOUND
          C
0047         10 IF(BOUNDR-BOUNDO)11,11,17                              S1500470
          C
          C NO.  USE THE NEW LOWER BOUND TO CONTINUE THE ALGORITHM
          C
0048         11 BOUNDO=BOUNDR                                          S1500480
0049            L(I)=LMAX(I)                                           S1500490
0050            DO 12 J=JX,IN                                          S1500500
0051         12 L(J)=LMIN(J)                                           S1500510
0052            GO TO 16                                               S1500520
          C
          C IS THE CURRENT LOWER BOUND SMALLER THAN THE NEW LOWER BOUND
          C
0053         13 IF(BOUNDS-BOUNDO)14,14,17                              S1500530
          C
          C NO.  USE THE NEW LOWER BOUND TO CONTINUE THE ALGORITHM
          C
0054         14 BOUNDO=BOUNDS                                          S1500540
0055            L(I)=LMIN(I)                                           S1500550
```

```
0056          DO 15 J=JX,IN                                               S1500560
0057       15 L(J)=LMAX(J)                                                S1500570
0058          TIMEAX=TIMEAY                                               S1500580
0059          TIMEBX=TIMEBY                                               S1500590
0060       16 CONTINUE                                                    S1500600
      C
      C THE CURRENT LOWER BOUND IS SMALLER THAN ANY OTHER LOWER BOUND
      C AN OPTIMAL SEQUENCE HAS BEEN FOUND
      C
0061       17 DO 20 I=1,IN                                                S1500610
0062          I1=L(I)                                                     S1500620
0063          GO TO (18,19),I1                                            S1500625
0064       18 AX(I)=A(I)                                                  S1500630
0065          BX(I)=B(I)                                                  S1500640
0066          GO TO 20                                                    S1500650
0067       19 AX(I)=B(I)                                                  S1500660
0068          BX(I)=A(I)                                                  S1500670
0069       20 CONTINUE                                                    S1500680
0070          RETURN                                                      S1500690
0071          END                                                         S1500700
```

SCALAR MAP

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
JN	128	IN	12C	I	130	TIMEAX	134	TIMEBX	138
TIMEAR	13C	TIMEBR	140	I1	144	BOUNDO	148	TIMEAY	14C
TIMEBY	150	TIMEAS	154	TIMEBS	158	JX	15C	J	160
BOUNDR	164	BOUNDS	168						

ARRAY MAP

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
A	16C	B	170	AX	174	BX	178	L	17C
LMAX	180	LMIN	194						

SUBPROGRAMS CALLED

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
AMIN1	1A8	AMAX1	1AC						

TOTAL MEMORY REQUIREMENTS 000916 BYTES

```
0001           SUBROUTINE PARAL2(A,B,C,D,AX,BX,L,IN)                               S1600010
          C  **************************************************************************
          C
          C SOLVE PARALLEL OPERATION SEQUENCING PROBLEM FOR SAME OPERS. DIFFERENT MACHINES
          C
          C **************************************************************************
0002           DIMENSION A(5),B(5),C(5),D(5),AX(5),BX(5),L(5),L1(5),LMAX(5)        S1600020
0003           DIMENSION LMIN(5)                                                   S1600030
0004           JN=IN-1                                                             S1600040
0005           DO 19 K=1,2                                                         S1600050
0006           DO 3 I=1,IN                                                         S1600060
0007           IF(A(I)-C(I))1,1,2                                                  S1600080
0008         1 LMAX(I)=2                                                           S1600100
0009           LMIN(I)=1                                                           S1600110
0010           GO TO 3                                                             S1600120
0011         2 LMAX(I)=1                                                           S1600130
0012           LMIN(I)=2                                                           S1600140
0013         3 CONTINUE                                                            S1600150
          C
          C IS THIS THE FIRST TREE PAIR OR THE SECOND TREE PAIR
          C
0014           GO TO (300,400),K                                                   S1600160
          C
          C FIRST PAIR.  CREATE A NODE FOR EACH TREE WITH FIRST OPERATION ON MACHINE A
          C
0015       300 TIMEAX=A(1)                                                         S1600170
0016           TIMEBX=D(1)                                                         S1600180
0017           TIMEAR=A(1)                                                         S1600190
0018           TIMEBR=D(1)                                                         S1600200
0019           GO TO 500                                                           S1600210
          C
          C SECOND PAIR.  CREATE A NODE FOR EACH TREE WITH SECOND OPERATION ON MACHINE A
          C
0020       400 TIMEAX=C(1)                                                         S1600220
0021           TIMEBX=B(1)                                                         S1600230
0022           TIMEAR=C(1)                                                         S1600240
0023           TIMEBR=B(1)                                                         S1600250
0024       500 L(1)=K                                                              S1600260
0025           I1=LMAX(1)                                                          S1600270
          C
          C WHCIH MACHINE HAS THE LARGER OPERATION TIME FOR THE FIRST JOB
          C
0026           GO TO (600,700),K                                                   S1600275
0027       600 GO TO (4,6),I1                                                      S1600280
0028       700 GO TO (6,4),I1                                                      S1600285
          C
          C MACHINE A.  ASSIGN SMALLER OPERATION TIMES OF REMAINING JOBS TO MACHINE A
          C
```

```
0029         4 DO 5 I=2,IN                                            S1600290
0030           TIMEAR=TIMEAR+AMIN1(A(I),C(I))                         S1600300
0031           TIMEBR=TIMEBR+AMAX1(B(I),D(I))                         S1600310
0032         5 L(I)=LMIN(I)                                           S1600320
0033           GO TO 8                                                S1600330
       C
       C MACHINE B.  ASSIGN SMALLER OPERATION TIMES OF REMAINING JOBS TO MAHCINE B
       C
0034         6 DO 7 I=2,IN                                            S1600340
0035           TIMEAR=TIMEAR+AMAX1(A(I),C(I))                         S1600350
0036           TIMEBR=TIMEBR+AMIN1(B(I),D(I))                         S1600360
0037         7 L(I)=LMAX(I)                                           S1600370
       C
       C COMPUTE THE LOWER BOUND FOR THE SEQUENCES CREATED
       C
0038         8 BOUNDO=AMAX1(TIMEAR,TIMEBR)                            S1600380
0039           DO 16 I=2,JN                                           S1600390
       C
       C EXTEND THE TREES BY ADDING BOTH POSSIBLE OPERATIONS TO BOTH TREES
       C
0040           TIMEAY=TIMEAX+AMIN1(A(I),C(I))                         S1600400
0041           TIMEBY=TIMEBX+AMAX1(B(I),D(I))                         S1600410
0042           TIMEAX=TIMEAX+AMAX1(A(I),C(I))                         S1600420
0043           TIMEBX=TIMEBX+AMIN1(B(I),D(I))                         S1600430
0044           TIMEAR=TIMEAX                                          S1600440
0045           TIMEBR=TIMEBX                                          S1600450
0046           TIMEAS=TIMEAY                                          S1600460
0047           TIMEBS=TIMEBY                                          S1600470
0048           JX=I+1                                                 S1600480
0049           DO 9 J=JX,IN                                           S1600490
0050           TIMEAR=TIMEAR+AMIN1(A(J),C(J))                         S1600500
0051           TIMEBR=TIMEBR+AMAX1(B(J),D(J))                         S1600510
0052           TIMEAS=TIMEAS+AMAX1(A(J),C(J))                         S1600520
0053         9 TIMEBS=TIMEBS+AMIN1(B(J),D(J))                         S1600530
       C
       C COMPUTE THE LOWER BOUNDS FOR THE NEW SEQUENCES
       C
0054           BOUNDR=AMAX1(TIMEAR,TIMEBR)                            S1600540
0055           BOUNDS=AMAX1(TIMEAS,TIMEBS)                            S1600550
       C
       C THE SMALLER OF THESE TWO LOWER BOUNDS IS THE NEW LOWER BOUND
       C
0056           IF(BOUNDR-BOUNDS)10,10,13                              S1600560
       C
       C IS THE CURRENT LOWER BOUND SMALLER THAN THE NEW LOWER BOUND
       C
0057        10 IF(BOUNDR-BOUNDO)11,11,17                              S1600570
       C
```

```
C NO.  USE THE NEW LOWER BOUND TO CONTINUE THE ALGORITHM
C
   11 BOUNDO=BOUNDR                                                S1600580
      L(I)=LMAX(I)                                                 S1600590
      DO 12 J=JX,IN                                                S1600600
   12 L(J)=LMIN(J)                                                 S1600610
      GO TO 16                                                     S1600620
C
C IS THE CURRENT LOWER BOUND SMALLER THAN THE NEW LOWER BOUND
C
   13 IF(BOUNDS-BOUNDO)14,14,17                                    S1600630
C
C NO.  USE THE NEW LOWER BOUND TO CONTINUE THE ALGORITHM
C
   14 BOUNDO=BOUNDS                                                S1600640
      L(I)=LMIN(I)                                                 S1600650
      DO 15 J=JX,IN                                                S1600660
   15 L(J)=LMAX(J)                                                 S1600670
      TIMEAX=TIMEAY                                                S1600680
      TIMEBX=TIMEBY                                                S1600690
   16 CONTINUE                                                     S1600700
C
C IS THIS THE FIRST TREE PAIR OR THE SECOND TREE PAIR
C
   17 GO TO (18,19),K                                              S1600710
C
C FIRST PAIR.  THE CURRENT LOWER BOUND IS SMALLER THAN ANY OTHER LOWER BOUND FOR
C THIS TREE PAIR.  SAVE THE SEQUENCE AND THE LOWER BOUND
C
   18 BOUND1=BOUNDO                                                S1600720
      DO 180 I=1,IN                                                S1600730
  180 L1(I)=L(I)                                                   S1600740
   19 CONTINUE                                                     S1600750
C
C SECOND PAIR. THE CURRENT LOWER BOUND IS SMALLER THAN ANY OTHER LOWER BOUND FOR
C THIS TREE PAIR.  THE SEQUENCE FOR THE TREE PAIR WHICH HAS THE SMALLER LOWER
C BOUND IS AN OPTIMAL SEQUENCE
C
      IF(BOUND1-BOUNDO)20,20,22                                    S1600760
   20 DO 21 I=1,IN                                                 S1600770
   21 L(I)=L1(I)                                                   S1600780
   22 DO 25 I=1,IN                                                 S1600790
      I1=L(I)                                                      S1600800
      GO TO (23,24),I1                                             S1600805
   23 AX(I)=A(I)                                                   S1600810
      BX(I)=D(I)                                                   S1600820
      GO TO 25                                                     S1600830
   24 AX(I)=C(I)                                                   S1600840
```

```
0086            BX(I)=B(I)                                              S1600850
0087         25 CONTINUE                                                S1600860
0088            RETURN                                                  S1600870
0089            END                                                     S1600880
```

SCALAR MAP

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
JN	17C	IN	180	K	184	I	188	TIMEAX	18C
TIMEBX	190	TIMEAR	194	TIMEBR	198	I1	19C	BOUNDO	1A0
TIMEAY	1A4	TIMEBY	1A8	TIMEAS	1AC	TIMEBS	1B0	JX	1B4
J	1B8	BOUNDR	1BC	BOUNDS	1C0	BOUND1	1C4		

ARRAY MAP

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
A	1C8	B	1CC	C	1D0	D	1D4	AX	1D8
BX	1DC	L	1E0	L1	1E4	LMAX	1F8	LMIN	20C

SUBPROGRAMS CALLED

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
AMIN1	220	AMAX1	224						

TOTAL MEMORY REQUIREMENTS 000CE6 BYTES

```
0001          SUBROUTINE TIME23(FRACM1,FRACB1,FRACC1,FRACM2,FRACB2,FRACC2,       S1700010
             1FRACM3,FRACB3,FRACC3,FRACMM,FRACMB,FRACMC,FRACBM,FRACBB,FRACBC,    S1700020
             2FRACCM,FRACCB,FRACCC,I,CPU,TIMEB,TIMEC)                            S1700030
        C ***********************************************************************
        C
        C CALCULATE MULTIPROGRAMMING TIMES FOR SECOND- AND THIRD-PRIORITY PROGRAMS
        C
        C ***********************************************************************
0002          INTEGER CPU                                                        S1700040
0003          DIMENSION FRACM1(12,2),FRACB1(12,2),FRACC1(12,2),FRACM2(12,2)      S1700050
0004          DIMENSION FRACB2(12,2),FRACC2(12,2),FRACM3(12,2),FRACB3(12,2)      S1700060
0005          DIMENSION FRACC3(12,2),FRACMM(11,2),FRACMB(11,2),FRACMC(11,2)      S1700070
0006          DIMENSION FRACBM(11,2),FRACBB(11,2),FRACBC(11,2),FRACCM(11,2)      S1700080
0007          DIMENSION FRACCB(11,2),FRACCC(11,2)                                S1700090
0008          TIMEC=0.                                                           S1700100
        C
        C DO WE HAVE TO CALCULATE THE TIME FOR THE SECOND PRIORITY PROGRAM
        C
0009          GO TO (1,4,1),I                                                    S1700110
        C
        C YES. CALCULATE THE TIME FOR PORTION OF PROGRAM OVERLAPPED WITH MPX CHANNEL OPS
        C
0010        1 TIMEB=0.                                                           S1700120
0011          CALL TIME2 (FRACM1,FRACM2,FRACMM,CPU,TIMEB)                        S1700130
0012          CALL TIME2 (FRACM1,FRACB2,FRACMB,CPU,TIMEB)                        S1700140
0013          CALL TIME2 (FRACB1,FRACM2,FRACBM,CPU,TIMEB)                        S1700150
0014          CALL TIME2 (FRACB1,FRACB2,FRACBB,CPU,TIMEB)                        S1700160
        C
        C ARE BURST MODE MULTIPLEXOR OPERATIONS OVERLAPPED WITH INSTRUCTION EXECUTION
        C
0015          IF(CPU-2040)2,2,3                                                  S1700170
        C
        C NO.  CALCULATE TIME FOR PORTION OF PROGRAM NOT OVERLAPPED WITH BURST MODE OPS
        C
0016        2 CALL TIME2 (FRACM1,FRACC2,FRACMC,CPU,TIMEB)                        S1700180
0017          CALL TIME2 (FRACB1,FRACC2,FRACBC,CPU,TIMEB)                        S1700190
0018          CALL TIME2 (FRACC1,FRACM2,FRACCM,CPU,TIMEB)                        S1700200
0019          CALL TIME2 (FRACC1,FRACB2,FRACCB,CPU,TIMEB)                        S1700210
0020          CALL TIME2 (FRACC1,FRACC2,FRACCC,CPU,TIMEB)                        S1700220
        C
        C DO WE HAVE TO CALCULATE THE TIME FOR THE THIRD PRIORITY PROGRAM
        C
0021        3 GO TO (6,4,4),I                                                    S1700230
        C
        C YES.  CALCULATE THE TIME FOR THE THIRD PRIORITY PROGRAM
        C
0022        4 CALL TIME3 (FRACM1,FRACM2,FRACM3,FRACMM,CPU,TIMEB,TIMEC)           S1700240
0023          CALL TIME3 (FRACM1,FRACM2,FRACB3,FRACMM,CPU,TIMEB,TIMEC)           S1700250
```

```
0024              CALL TIME3 (FRACM1,FRACB2,FRACM3,FRACMB,CPU,TIMEB,TIMEC)           S1700260
0025              CALL TIME3 (FRACM1,FRACB2,FRACB3,FRACMB,CPU,TIMEB,TIMEC)           S1700270
0026              CALL TIME3 (FRACB1,FRACM2,FRACM3,FRACBM,CPU,TIMEB,TIMEC)           S1700280
0027              CALL TIME3 (FRACB1,FRACM2,FRACB3,FRACBM,CPU,TIMEB,TIMEC)           S1700290
0028              CALL TIME3 (FRACB1,FRACB2,FRACM3,FRACBB,CPU,TIMEB,TIMEC)           S1700300
0029              CALL TIME3 (FRACB1,FRACB2,FRACB3,FRACBB,CPU,TIMEB,TIMEC)           S1700310
            C
            C ARE BURST MODE MULTIPLEXOR OPERATIONS OVERLAPPED WITH INSTRUCTION EXECUTION
            C
0030              IF(CPU-2040)5,5,6                                                  S1700320
            C
            C NU.  CALCULATE TIME FOR PORTION OF PROGRAM NOT OVERLAPPED WITH BURST MODE OPS
            C
0031            5 CALL TIME3 (FRACM1,FRACM2,FRACC3,FRACMM,CPU,TIMEB,TIMEC)           S1700330
0032              CALL TIME3 (FRACM1,FRACB2,FRACC3,FRACMB,CPU,TIMEB,TIMEC)           S1700340
0033              CALL TIME3 (FRACB1,FRACM2,FRACC3,FRACBM,CPU,TIMEB,TIMEC)           S1700350
0034              CALL TIME3 (FRACB1,FRACB2,FRACC3,FRACBB,CPU,TIMEB,TIMEC)           S1700360
0035              CALL TIME3 (FRACM1,FRACC2,FRACM3,FRACMC,CPU,TIMEB,TIMEC)           S1700370
0036              CALL TIME3 (FRACM1,FRACC2,FRACB3,FRACMC,CPU,TIMEB,TIMEC)           S1700380
0037              CALL TIME3 (FRACM1,FRACC2,FRACC3,FRACMC,CPU,TIMEB,TIMEC)           S1700390
0038              CALL TIME3 (FRACB1,FRACC2,FRACM3,FRACBC,CPU,TIMEB,TIMEC)           S1700400
0039              CALL TIME3 (FRACB1,FRACC2,FRACB3,FRACBC,CPU,TIMEB,TIMEC)           S1700410
0040              CALL TIME3 (FRACB1,FRACC2,FRACC3,FRACBC,CPU,TIMEB,TIMEC)           S1700420
0041              CALL TIME3 (FRACC1,FRACM2,FRACM3,FRACCM,CPU,TIMEB,TIMEC)           S1700430
0042              CALL TIME3 (FRACC1,FRACM2,FRACB3,FRACCM,CPU,TIMEB,TIMEC)           S1700440
0043              CALL TIME3 (FRACC1,FRACM2,FRACC3,FRACCM,CPU,TIMEB,TIMEC)           S1700450
0044              CALL TIME3 (FRACC1,FRACB2,FRACM3,FRACCB,CPU,TIMEB,TIMEC)           S1700460
0045              CALL TIME3 (FRACC1,FRACB2,FRACB3,FRACCB,CPU,TIMEB,TIMEC)           S1700470
0046              CALL TIME3 (FRACC1,FRACB2,FRACC3,FRACCB,CPU,TIMEB,TIMEC)           S1700480
0047              CALL TIME3 (FRACC1,FRACC2,FRACM3,FRACCC,CPU,TIMEB,TIMEC)           S1700490
0048              CALL TIME3 (FRACC1,FRACC2,FRACB3,FRACCC,CPU,TIMEB,TIMEC)           S1700500
0049              CALL TIME3 (FRACC1,FRACC2,FRACC3,FRACCC,CPU,TIMEB,TIMEC)           S1700510
0050            6 RETURN                                                             S1700520
0051              END                                                                S1700530
```

```
0001          SUBROUTINE ERRPRT(ERROR)                                          S1800010
        C **********************************************************************
        C
        C PRINT ERROR HEADING AND UPDATE ERROR COUNT
        C
        C **********************************************************************
0002          INTEGER ERROR                                                     S1800020
0003          ERROR=ERROR+1                                                     S1800030
0004          WRITE (6,1)                                                       S1800040
0005        1 FORMAT(28H0********** ERROR **********)                           S1800050
0006          RETURN                                                            S1800060
0007          END                                                               S1800070
```

BIBLIOGRAPHY

Codd, E. F., "Multiprogram Scheduling: Parts 1 and 2. Introduction and Theory," Communications of the ACM, III (June, 1960), 347-350.

Codd, E. F., "Multiprogram Scheduling: Parts 3 and 4. Scheduling Algorithm and External Constraints," Communications of the ACM, III (July, 1960), 413-418.

Heller, J., "Sequencing Aspects of Multiprogramming," Journal of the Association for Computing Machinery, VIII (July, 1961), 426-439.

Hutchinson, George K. "A Computer Center Simulation Project," Communications of the ACM, VIII (September, 1965), 559-568.

IBM System/360 Principles of Operation. Poughkeepsie: IBM Corporation, 1964.

IBM System/360 Disk Operating System: System Control and Service Programs. Endicott: IBM Corporation, 1966.

Ignall, Edward and Schrage, Linus. "Application of the Branch and Bound Technique to Some Flow-Shop Scheduling Problems," Operations Research, XIII (May-June, 1965), 400-412.

Johnson, S.M., "Optimal Two- and Three-Stage Production Schedules with Setup Times Included," Chapter 2 of Muth, John F. and Thompson, Gerald L., editors. Industrial Scheduling. Englewood Cliffs: Prentice-Hall, 1963, 13-20.

Little, John D. C., Murty, Katta G., Sweeney, Dura W., and Karel, Caroline. "An Algorithm for the Traveling Salesman Problem," Operations Research, XI (November-December, 1963), 972-989.

Lomnicki, Z. A., "A Branch- and- Bound Algorithm for the Exact Solution of the Three-machine Scheduling Problem," Operations Research Quarterly, XVI (March, 1965), 89-100.

Lynch, W. C., "Description of a High Capacity, Fast Turnaround University Computing Center," Communications of the ACM, IX (February, 1966), 117-123.

McKenney, James L., Simultaneous Processing of Jobs on an Electronic Computer," Management Science VIII (April, 1962), 344-354.

Schwartz, Eugene S., "An Automatic Sequencing Procedure with Application to Parallel Programming," Journal of the Association for Computing Machinery, VIII (October, 1961), 513-537.